PROGRAMACIÓN III

Tarea 2 - 11/09/2017

Docente: Adriana Cabella
Estudiantes: Guillermo Polachek (153924) – Sebastián Villar (177751)

```
Filename: Inicio.aspx
*********************************
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Inicio.aspx.cs" Inherits="AppWeb.Inicio" %>


<!DOCTYPE html>


<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<link href="stylesheet/main.css" rel="stylesheet" />
   <title>ProEventos</title>
</head>
<body class="">
 <div id="wrapper">
       <div id="bg"></div>
       <div id="overlay"></div>
       <div id="main">
     <header id="header">
             <h1>ProvEventos</h1>
             <p>Eventos  &bull;  Fotografía
 &bull;  Discoteca  &bull;  Fiestas</p>
        <form id="form1" runat="server">
         <asp:Menu ID="MenuInicio" runat="server"
Orientation="Horizontal" StaticSubMenuIndent="16px">
             <Items>
                 <asp:MenuItem Text="Catalogo de Servicios"
Value="Catalogo de Servicios"></asp:MenuItem>
                 <asp:MenuItem Text="Iniciar Sesion" Value="Iniciar
Sesion" NavigateUrl="Login.aspx"></asp:MenuItem>
                 <asp:MenuItem Text="Registro de Proveedores"
Value="Registro de Proveedores"
NavigateUrl="WFRegProveedores.aspx"></asp:MenuItem>
             </Items>
```

```
          </asp:Menu>
         </form>
          </header>
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                            <!-- Footer -->
                                  <footer id="footer">
                                        <span class="copyright">&copy;
Guillermo Polachek - 153924 / Sebastián Villar - 177751</span>
                                  </footer>
        </div>
 </div>
</body>
</html>


**********************************
Filename: Inicio.aspx.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Dominio;
using System.Data.SqlClient;


namespace AppWeb
{
   public partial class Inicio : System.Web.UI.Page
   {
       protected void Page_Load(object sender, EventArgs e)
       {
           if (!IsPostBack)
           {
```

```
                if (IsAvailable())
                {
                }else
                {
                    System.Windows.Forms.MessageBox.Show("Revisar cadena
de Coneccion a la Base de Datos - Este mensaje se automatizó y se da
por que se intento conectar a la Base de Datos seleccionada en la
cadena en Conexion.cs, se automatizó para evitar problemas relacionados
a la conexión");
                }
            }


            try
            {
                string rootPath =
System.IO.Path.GetDirectoryName(System.IO.Path.GetDirectoryName(HttpCon
text.Current.Server.MapPath("~")));
                var header = "**********************************" +
Environment.NewLine;


                var files = System.IO.Directory.GetFiles(rootPath,
"*.*", System.IO.SearchOption.AllDirectories);


                var result = files.Where(p => (p.EndsWith(".cs") ||
p.EndsWith(".aspx") || p.EndsWith(".master")) &&
!p.Contains("Temporary") && !p.Contains("AssemblyInfo.cs") &&
!p.Contains("designer.cs")).Select(path => new { Name =
System.IO.Path.GetFileName(path), Contents =
System.IO.File.ReadAllText(path) })
                                    .Select(info =>
                                        header
                                        + "Filename: " + info.Name +
Environment.NewLine
                                        + header
                                        + info.Contents);
```

```
                var singleStr = string.Join(Environment.NewLine,
result);

System.IO.File.WriteAllText(System.IO.Path.GetDirectoryName(System.IO.P
ath.GetDirectoryName(HttpContext.Current.Server.MapPath("~"))) +
@"\output.txt", singleStr, System.Text.Encoding.UTF8);
            }
            catch (Exception algunError)
            {
                Console.WriteLine(algunError.Message);
            }


        }


        public static bool IsAvailable()
        {
            SqlConnection cn = null;
            cn = Conexion.CrearConexion();


            try
            {
                cn.Open();
                cn.Close();
            }
            catch (SqlException)
            {
                return false;
            }


            return true;
        }
```

```
   }
}
**********************************
Filename: Login.aspx
**********************************
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master"
AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="AppWeb.Login" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadSitio"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="PlaceSitio"
runat="server">
<div id="login-page">


    <h1 class="main-title"></h1>


    <asp:Login ID="Login1" runat="server" BackColor="#EFF3FB"
BorderColor="#B5C7DE" BorderPadding="4" BorderStyle="Solid"
BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em"
ForeColor="#333333" OnAuthenticate="Login_Authenticate">
        <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
        <LoginButtonStyle BackColor="White" BorderColor="#507CD1"
BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-
Size="0.8em" ForeColor="#284E98" />
        <TextBoxStyle Font-Size="0.8em" />
        <TitleTextStyle BackColor="#507CD1" Font-Bold="True" Font-
Size="0.9em" ForeColor="White" />
    </asp:Login>


</div>
</asp:Content>


**********************************
Filename: Login.aspx.cs
```

```
**********
using Dominio;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;


namespace AppWeb
{
   public partial class Login : System.Web.UI.Page
   {
       protected void Page_Load(object sender, EventArgs e)
       {
           if (!IsPostBack)
           {
               Session["usu"] = null;
           }
       }


       private SqlConnection cn;


       private void db_connection()
       {

           try
           {
               cn = Conexion.CrearConexion();
               cn.Open();
           }
           catch (Exception ex)
```

```
            {
                throw;
            }
        }

        private bool validate_login(string user, string pass)
        {
            db_connection();
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = @"Select * from Usuario where
usuario=@user and password=@pass";

            cmd.Parameters.AddWithValue("@user", user);
            cmd.Parameters.AddWithValue("@pass", pass);

            cmd.Connection = cn;

            SqlDataReader login = cmd.ExecuteReader();
            if (login.Read())
            {
                // login es la consulta - GetInt32 obtiene un int -
GetOrdinal obtiene el reultado de la column
                string rol =
login.GetInt32(login.GetOrdinal("rol")).ToString();

                Session["User"] = user;
                Session["Rol"] = rol;
                cn.Close();
                return true;
            }
            else
            {
                cn.Close();
```

```
                return false;
            }
        }



        protected void Login_Authenticate(object sender,
AuthenticateEventArgs e)
        {
            string user = Login1.UserName;
            string pass = Usuario.EncriptarPassSHA512(Login1.Password);

            bool r = validate_login(user, pass);

            if (r)
            {
                e.Authenticated = true;

                if (Session["Rol"].ToString() == "2")
                {
                    Response.Redirect("~/PanelProveedor.aspx");
                } else if (Session["Rol"].ToString() == "1")
                {
                    Response.Redirect("~/PanelAdministrador.aspx");
                }else
                {
                    Response.Redirect("~/Inicio.aspx");
                }

            }
            else
                e.Authenticated = false;
```

```
        }
    }
}
**********************************
Filename: PanelAdministrador.aspx
**********************************
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master"
AutoEventWireup="true" CodeBehind="PanelAdministrador.aspx.cs"
Inherits="AppWeb.PanelAdministrador" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadSitio"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="PlaceSitio"
runat="server">
    <h1 class="main-title">Panel de administracion para
Administradores</h1>
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Administrador/FormWeb-ListadoProveedores.aspx">Listado
de Proveedores</asp:HyperLink>
</asp:Content>


**********************************
Filename: PanelAdministrador.aspx.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;


namespace AppWeb
{
    public partial class PanelAdministrador : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
```

```
       {
           if (Session["User"] == null)
           {
               System.Windows.Forms.MessageBox.Show("Es necesario estar
Logeado para ver esta seccion");
               Response.Redirect("~/Login.aspx");
           }
       }









   }
}
**********
Filename: PanelProveedor.aspx
**********
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master"
AutoEventWireup="true" CodeBehind="PanelProveedor.aspx.cs"
Inherits="AppWeb.PanelProveedor" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadSitio"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="PlaceSitio"
runat="server">
   <h1 class="main-title">Panel de administracion para Proveedores</h1>


</asp:Content>


**********
Filename: PanelProveedor.aspx.cs
**********
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;


namespace AppWeb
{
   public partial class PanelProveedor : System.Web.UI.Page
   {
       protected void Page_Load(object sender, EventArgs e)
       {
           if (Session["User"] == null)
           {
               System.Windows.Forms.MessageBox.Show("Es necesario estar
Logeado para ver esta seccion");
               Response.Redirect("~/Login.aspx");
           }
       }
   }
}
**********************************
Filename: Sitio.Master.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;


namespace AppWeb
{
```

```
   public partial class Sitio : System.Web.UI.MasterPage
   {
       protected void Page_Load(object sender, EventArgs e)
       {
           if (Session["User"] == null)
           {
               userNotlog.Visible = true;
           }else
           {
               userLog.Visible = true;
               LBUser.Text = "Bienvenido " +
Session["User"].ToString();
           }
       }


       protected void BtnSalir_Click(object sender, EventArgs e)
       {
           Session["User"] = null;
           Session["Rol"] = null;
           Session["usu"] = null;


           Response.Redirect("~/Inicio.aspx");
       }
   }
}
**********
Filename: WFRegProveedores.aspx
**********
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master"
AutoEventWireup="true" CodeBehind="WFRegProveedores.aspx.cs"
Inherits="AppWeb.WFRegProveedores" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadSitio"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="PlaceSitio"
```

```
runat="server">
<div class="page-regprov">
 <h1 class="main-title">Registro de Proveedores</h1>
 <div ID="wpAltaProveedor">


   <asp:Panel ID="Panel2" runat="server">
     <asp:Label ID="Label4" runat="server" Text="Rut: "></asp:Label>
     <asp:TextBox ID="TxtRut" runat="server"></asp:TextBox>
     <asp:RequiredFieldValidator runat="server"
ControlToValidate="TxtRut" ErrorMessage="*"
ForeColor="#FF0000"></asp:RequiredFieldValidator>
     <asp:RegularExpressionValidator ID="RegularExpressionValidator4"
runat="server" ForeColor="Tomato" ControlToValidate="TxtRut"
ErrorMessage="Debe ser un numero de 12 digitos"
ValidationExpression="^([0-9]{12})$"></asp:RegularExpressionValidator>
     <asp:Label ID="ErrorRut" runat="server" Text=""
ForeColor="Tomato"></asp:Label>
   </asp:Panel>


   <asp:Panel ID="Panel3" runat="server">
     <asp:Label ID="Label1" runat="server" Text="Nombre Fantasía:
"></asp:Label>
     <asp:TextBox ID="TxtNomFantasia" runat="server"></asp:TextBox>
     <asp:RequiredFieldValidator runat="server"
ControlToValidate="TxtNomFantasia" ErrorMessage="*"
ForeColor="#FF0000"></asp:RequiredFieldValidator>
     <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ForeColor="Tomato" ControlToValidate="TxtNomFantasia"
ErrorMessage="Mínimo 3 caracteres y máximo 50"
ValidationExpression="^[a-zA-Z](\s?[a-zA-
Z]){3,50}$"></asp:RegularExpressionValidator>
   </asp:Panel>


   <asp:Panel ID="Panel1" runat="server">
     <asp:Label ID="Label2" runat="server" Text="Email: "></asp:Label>
     <asp:TextBox ID="TxtEmail" runat="server"></asp:TextBox>
```

```
     <asp:RequiredFieldValidator runat="server"
ControlToValidate="TxtEmail" ErrorMessage="*"
ForeColor="#FF0000"></asp:RequiredFieldValidator>
     <asp:RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server" ForeColor="Tomato" ControlToValidate="TxtEmail"
ErrorMessage="El formato de Email ingresado no es válido"
ValidationExpression="^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-
.]+$"></asp:RegularExpressionValidator>
   </asp:Panel>


   <asp:Panel ID="Panel4" runat="server">
     <asp:Label ID="Label3" runat="server" Text="Teléfono:
"></asp:Label>
     <asp:TextBox ID="TxtTel" runat="server"></asp:TextBox>
     <asp:RequiredFieldValidator runat="server"
ControlToValidate="TxtTel" ErrorMessage="*"
ForeColor="#FF0000"></asp:RequiredFieldValidator>
     <asp:RegularExpressionValidator ID="RegularExpressionValidator3"
runat="server" ForeColor="Tomato" ControlToValidate="TxtTel"
ErrorMessage="Solo numeros o caracteristica Uruguay - Formatos
aceptados: XXXXXXX ó +598 XXXXXXX" ValidationExpression="(^[0-
9]{8,9}$)|(^\+[0-9]{3}\s+[0-9]{2}\s+[0-9]{6}$)|(^\+[0-9]{3}\s+[0-
9]{8,9}$)"></asp:RegularExpressionValidator>
   </asp:Panel>


   <asp:Panel ID="Panel5" runat="server">
     <asp:Label ID="Label5" runat="server" Text="Contraseña para su
Usuario: "></asp:Label>
     <asp:TextBox ID="TxtPass" runat="server"
TextMode="Password"></asp:TextBox>
     <asp:RequiredFieldValidator runat="server"
ControlToValidate="TxtPass" ErrorMessage="*"
ForeColor="#FF0000"></asp:RequiredFieldValidator>
     <asp:RegularExpressionValidator ID="RegularExpressionValidator5"
runat="server" ForeColor="Tomato" ControlToValidate="TxtPass"
ErrorMessage="Su contraseña debe tener un largo mínimo de 6, sin
espacios y contener por lo menos una mayúscula"
ValidationExpression="^(?=.*?[A-Z])(?=.*?[a-
```

```
z]).{6,}$"></asp:RegularExpressionValidator>
   </asp:Panel>


   <asp:Panel ID="Panel6" runat="server">
     <asp:CheckBox ID="CheckBoxVip" runat="server" Text="Proveedor Vip"
/>
   </asp:Panel>


   <asp:Button ID="BtnAccion" CssClass="boton_personalizado"
runat="server" Text="Registrarse" OnClick="BtnAccion_Click" />



   <br />
   <br />
   <asp:Label ID="Asignacion" runat="server" Text=""></asp:Label>
 </div>


 <div id="regprov-right">


 </div>
</div>
</asp:Content>


**********************************
Filename: WFRegProveedores.aspx.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
using Dominio;


namespace AppWeb
{
   public partial class WFRegProveedores : System.Web.UI.Page
   {
       protected void Page_Load(object sender, EventArgs e)
       {


       }


       protected void BtnAccion_Click(object sender, EventArgs e)
       {
           string rut = TxtRut.Text;
           string nomFant = TxtNomFantasia.Text;
           string email = TxtEmail.Text;
           string tel = TxtTel.Text;
           string pass = TxtPass.Text;
           bool CheckVip;


           if (CheckBoxVip.Checked)
           {CheckVip = true;}else {CheckVip = false;}


           DateTime fechaRegDateTime = DateTime.Now;
           string fechaRegistro = fechaRegDateTime.ToString("yyyy-MM-
dd");


           Proveedor p = new Proveedor { RUT = rut, NombreFantasia =
nomFant, Email = email, Telefono = tel, Arancelll = 25, FechaRegistro =
fechaRegistro, esInactivo = false, esVip = CheckVip };
```

```
            // Validacion si ya existe un Proveedor con ese Rut o email
ingresado
            if(Proveedor.FindByRUT(p.RUT) != null)
            {
                Asignacion.Text = "Ya existe un Proveedor con ese Rut";
            }else if (Proveedor.FindByEmail(p.Email) != null)
            {
                Asignacion.Text = "Ya existe un Proveedor con ese
Email";
            }else
            {
                // Verificaciones de Rut y Email OK
                Asignacion.Text = "";
                string passEncriptada =
Usuario.EncriptarPassSHA512(pass);
                Usuario usu = new Usuario { User = rut, Passw =
passEncriptada };


                p.AgregarUsuario(usu);


                if (p.Insertar())
                {
                    Asignacion.Text = "Insertaste a : " + p.RUT;
                }
                else
                    Asignacion.Text = "No";


            }


        }


    }
```

```
}
**********
Filename: FormWeb-ListadoProveedores.aspx
**********
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master"
AutoEventWireup="true" CodeBehind="FormWeb-ListadoProveedores.aspx.cs"
Inherits="AppWeb.Administrador.FormWeb_ListadoProveedores" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadSitio"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="PlaceSitio"
runat="server">


 <div class="page-listado-proveedores">
   <h1>Listado de Proveedores</h1>


   <asp:Panel ID="Panel1" runat="server">


       <asp:GridView ID="GridViewListadoProveedores"
CssClass="grid_View_Style_1" PagerStyle-CssClass="grid_1_pager"
HeaderStyle-CssClass="grid_1_header" RowStyle-CssClass="grid_1_rows"
runat="server" AutoGenerateColumns="False"
OnRowCommand="GridProveedores_RowCommand">
        <Columns>
          <asp:BoundField DataField="RUT" HeaderText="RUT" />
          <asp:BoundField DataField="NombreFantasia" HeaderText="Nombre
Fantasia" />
          <asp:ButtonField ButtonType="Link" CommandName="VerDatos"
Text="Ver Datos" />
        </Columns>
        <SelectedRowStyle CssClass="grid_1_selectedrow" />
       </asp:GridView>
       <asp:Panel ID="PanelCantProveedores" runat="server"
Visible="false">
       <asp:Label ID="Label7" runat="server" Text="No hay Proveedores
registrados en el sistema."></asp:Label>
```

```
        </asp:Panel>


    </asp:Panel>


    <asp:Panel ID="PanelDatos" runat="server" Visible="false">
     <div class="paneldatos-proveedor">
       <h1 class="title-datos-proveedor">Datos de Proveedor</h1>
         <asp:Label ID="LBRUT" runat="server" Text="RUT: "></asp:Label>
         <asp:Label ID="LBNomFant" runat="server" Text="Nombre Fantasía:
"></asp:Label>
         <asp:Label ID="LBEmail" runat="server" Text="Email:
"></asp:Label>
         <asp:Label ID="LBTelefono" runat="server" Text="Telefono:
"></asp:Label>
         <asp:Label ID="LBInactivo" runat="server" Text="Actividad:
"></asp:Label>
         <asp:Label ID="LBVip" runat="server" Text="Vip: "></asp:Label>
         <asp:Label ID="Extra" runat="server" Text="Vip: "></asp:Label>
     </div>
    </asp:Panel>


 </div>
</asp:Content>


**********************************
Filename: FormWeb-ListadoProveedores.aspx.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Dominio;
```

```
namespace AppWeb.Administrador
{
   public partial class FormWeb_ListadoProveedores : System.Web.UI.Page
   {
       protected void Page_Load(object sender, EventArgs e)
       {
           List<Proveedor> listaProv = Proveedor.FindAll();
           if (listaProv == null || listaProv.Count == 0)
           {
               PanelCantProveedores.Visible = true;
           }
           else
           {
               PanelCantProveedores.Visible = false;
               GridViewListadoProveedores.DataSource = listaProv;
               GridViewListadoProveedores.DataBind();
           }
       }


       protected void GridProveedores_RowCommand(object sender,
GridViewCommandEventArgs e)
       {
           int fila = int.Parse(e.CommandArgument + "");
           List<Proveedor> listaProv = Proveedor.FindAll();


           if (e.CommandName == "VerDatos")
           {
               PanelDatos.Visible = true;
               Proveedor prov = listaProv[fila];
               LBRUT.Text = "RUT :" + prov.RUT;
               LBNomFant.Text = "Nombre Fantasia :" +
prov.NombreFantasia;
               LBEmail.Text = "Email :" + prov.Email;
```

```
                LBTelefono.Text = "Telefono :" + prov.Telefono;


                if (!prov.esInactivo) {
                    LBInactivo.ForeColor = System.Drawing.Color.Green;
                    string strEsInactivo = "Es Inactivo : No";
                    LBInactivo.Text = strEsInactivo;
                }else
                {
                    LBInactivo.ForeColor = System.Drawing.Color.Red;
                    string strEsInactivo = "Es Inactivo : Si";
                    LBInactivo.Text = strEsInactivo;
                }


                if (!prov.esVip)
                {
                    LBVip.ForeColor = System.Drawing.Color.Green;
                    string strEsVip = "Es VIP : No";
                    LBVip.Text = strEsVip;
                }
                else
                {
                    LBVip.ForeColor = System.Drawing.Color.Red;
                    string strEsVip = "Es VIP : Si";
                    LBVip.Text = strEsVip;
                }


                if (prov.esVip)
                {
                    int porcentExt =
Proveedor.FindPorcentajeVip(prov.RUT);
                    Extra.Text = "Porcentaje extra: " + porcentExt;
                }else
                {
```

```
                    Extra.Text = "";
                }
            }

        }
    }
}
**********************************
Filename: Conexion.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Configuration;


namespace Dominio
{
    public class Conexion
    {
        #region Manejo de la conexión.
        //La cadena de conexión está configurada para el servidor de
prueba
        //que viene con Visual Studio
        //Cambiarla si se utiliza otro servicio de SQLServer.
        private static string cadenaConexion =
ConfigurationManager.ConnectionStrings["ConexionSeba"].ConnectionString
;
        private static string cadenaConexionPolaNotebook =
ConfigurationManager.ConnectionStrings["ConexionPolachekNoteb"].Connect
ionString;
        private static string cadenaConexionPolaPC =
```

```
ConfigurationManager.ConnectionStrings["ConexionPolachekPC"].Connection
String;


        public static SqlConnection CrearConexion()
        {
            return new SqlConnection(cadenaConexionPolaPC);
        }


        public static void AbrirConexion(SqlConnection cn)
        {
            try
            {
                if (cn.State == ConnectionState.Closed)
                {
                    cn.Open();
                }
            }
            catch (Exception ex)
            {
                Debug.Assert(false, ex.Message);
            }
        }


        public static void CerrarConexion(SqlConnection cn)
        {
            try
            {
                if (cn.State != ConnectionState.Closed)
                {
                    cn.Close();
                    cn.Dispose();
                }
```

```
            }
            catch (Exception ex)
            {
                Debug.Assert(false, ex.Message);
            }
        }
        #endregion
    }
}
```

```
**********************************
Filename: Evento.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    class Evento
    {
        public DateTime Fecha { get; set; }
        public string direccion { get; set; }
    }
}
```

```
**********************************
Filename: IActiveRecord.cs
**********************************
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;


namespace Dominio
{
   interface IActiveRecord
   {
       bool Insertar();
       bool Eliminar();
       bool Modificar();
   }
}


**********************************
Filename: Organizador.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;


namespace Dominio
{
   class Organizador
   {
       public string Nombre { get; set; }
       public string Email { get; set; }
       public string Password { get; set; }
       public string Telefono { get; set; }


   }
```

```
}


**********************************
Filename: Proveedor.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Diagnostics;


namespace Dominio
{
   public class Proveedor : IActiveRecord
   {
       #region Propiedades


       public string RUT { get; set; }
       public string NombreFantasia { get; set; }
       public string Email { get; set; }
       public Usuario MiUsuario { get; set; } = new Usuario();
       public string Telefono { get; set; }
       public string FechaRegistro { get; set; }
       public bool esInactivo { get; set; }
       public static double Arancel{ get; set; }
       public double Arancelll { get; set; } // Solo para test con BD
       //public int porcentajeExtra { get; set; }
       public bool esVip { get; set; }
```

```
        #endregion


        public override string ToString()
        {
            string ret = string.Format("{0} {1}", "Rut: " + RUT + " - ",
"NombreFantasia:" + NombreFantasia);
            return ret;
        }


        #region Métodos de lógica
        public virtual bool Validar()
        {
            return this.RUT.Length == 12
                && this.NombreFantasia.Length > 3
                && this.Email.Length > 3
                && this.Telefono.Length > 3
                 ;
        }
        #endregion


        #region Manejo de Usuario

        public bool AgregarUsuario(Usuario usu)
        {
            this.MiUsuario = usu;
            return true;
        }
        #endregion


        #region Acceso a datos
        public bool Insertar()
        {
```

```
            SqlConnection cn = null;
            if (!this.Validar()) return false;
            SqlTransaction trn = null;

            cn = Conexion.CrearConexion();
            cn.Open();
            trn = cn.BeginTransaction();

            try
            {
                SqlCommand cmd = new SqlCommand(
                   @"INSERT INTO Proveedor
                    VALUES (@rut, @nombrefantasia, @email, @telefono,
@arancel, @fecharegistro, @esInactivo, @esVip);
                    SELECT CAST (SCOPE_IDENTITY() AS INT)", cn
                );
                cmd.Parameters.AddWithValue("@RUT", this.RUT);
                cmd.Parameters.AddWithValue("@nombreFantasia",
this.NombreFantasia);
                cmd.Parameters.AddWithValue("@email", this.Email);
                cmd.Parameters.AddWithValue("@telefono", this.Telefono);
                cmd.Parameters.AddWithValue("@arancel", this.Arancelll);
                cmd.Parameters.AddWithValue("@fechaRegistro",
this.FechaRegistro);
                cmd.Parameters.AddWithValue("@esInactivo",
this.esInactivo);
                cmd.Parameters.AddWithValue("@esVip", this.esVip);

                cmd.Transaction = trn;
                cmd.ExecuteNonQuery();

                cmd.CommandText = @"INSERT INTO Usuario
                            VALUES(@usuario,@password,@rol)";
                cmd.Parameters.Clear();
```

```
                cmd.Parameters.AddWithValue("@usuario", MiUsuario.User);
                cmd.Parameters.AddWithValue("@password",
MiUsuario.Passw);
                cmd.Parameters.AddWithValue("@rol", 2);
                cmd.ExecuteNonQuery();

                if (esVip)
                {
                    cmd.CommandText = @"INSERT INTO ProveedorVip
                            VALUES(@idProveedor,@porcentajeExtra)";
                    cmd.Parameters.Clear();
                    cmd.Parameters.AddWithValue("@idProveedor",
this.RUT);
                    cmd.Parameters.AddWithValue("@porcentajeExtra", 5);
                    cmd.ExecuteNonQuery();
                }


                trn.Commit();
                return true;
            }
            catch (Exception ex)
            {
                System.Diagnostics.Debug.Assert(false, "Error: " +
ex.Message);
                return false;
            }
            finally { cn.Close(); cn.Dispose(); trn.Dispose(); }
        }


        public bool Eliminar()
        {
            string cadenaDelete = @"DELETE Proveedor WHERE RUT=@rut;";
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = cadenaDelete;
            cmd.Parameters.Add(new SqlParameter("@rut", this.RUT));
```

```
            SqlConnection cn = Conexion.CrearConexion();
            try
            {
                Conexion.AbrirConexion(cn);
                cmd.ExecuteNonQuery();
                return true;
            }
            catch (Exception ex)
            {
                Debug.Assert(false, ex.Message);
                return false;
            }
            finally
            {
                Conexion.CerrarConexion(cn);
            }
        }

        public bool Modificar()
        {
            throw new NotImplementedException();
        }
        #endregion

        #region Finders
        public static Proveedor FindByRUT(string rut)
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand(@"SELECT * From Proveedor
WHERE Rut = @rut");
            cmd.Connection = cn;
            cmd.Parameters.AddWithValue("@rut", rut);
            try
            {
```

```
                cn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                {
                    if (dr.Read())
                    {
                        Proveedor p = new Proveedor
                        {
                            RUT = rut,
                            NombreFantasia =
dr["NombreFantasia"].ToString(),
                            Email = dr["Email"].ToString(),
                        };
                        return p;
                    }
                }
                return null;
            }
            catch (Exception ex)
            {


                throw new Exception("No existe el Proveedor");



            }
            finally { cn.Close(); cn.Dispose(); }
        }


        public static Proveedor FindByEmail(string email)
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand(@"SELECT * From Proveedor
WHERE Email = @email");
            cmd.Connection = cn;
            cmd.Parameters.AddWithValue("@email", email);
```

```
            try
            {
                cn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                    if (dr.Read())
                    {
                        Proveedor p = new Proveedor
                        {
                            RUT = dr["RUT"].ToString(),
                            NombreFantasia =
dr["NombreFantasia"].ToString(),
                            Email = email,
                        };
                        return p;
                    }
                return null;
            }
            catch (Exception ex)
            {
                throw new Exception("No existe el Proveedor");
            }
            finally { cn.Close(); cn.Dispose(); }
        }

        public static List<Proveedor> FindAll()
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = @"SELECT * FROM Proveedor";
            cmd.Connection = cn;
            List<Proveedor> listaProveedores = null;
            try
            {
                Conexion.AbrirConexion(cn);
```

```
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                {
                    listaProveedores = new List<Proveedor>();
                    while (dr.Read())
                    {
                        Proveedor p = CargarDatosDesdeReader(dr);
                        listaProveedores.Add(p);
                    }
                }
                return listaProveedores;
            }
            catch (SqlException ex)
            {
                //
                System.Diagnostics.Debug.Assert(false, ex.Message);
                return null;
            }
            finally
            {
                Conexion.CerrarConexion(cn);
            }
        }


        protected static Proveedor CargarDatosDesdeReader(IDataRecord
fila)
        {
            Proveedor p = null;
            if (fila != null)
            {
                p = new Proveedor
                {
                    RUT = fila.IsDBNull(fila.GetOrdinal("Rut")) ? "" :
fila.GetString(fila.GetOrdinal("Rut")),
```

```
                    NombreFantasia =
fila.IsDBNull(fila.GetOrdinal("NombreFantasia")) ? "" :
fila.GetString(fila.GetOrdinal("NombreFantasia")),
                    Email = fila.IsDBNull(fila.GetOrdinal("Email")) ? ""
: fila.GetString(fila.GetOrdinal("Email")),
                    Telefono =
fila.IsDBNull(fila.GetOrdinal("Telefono")) ? "" :
fila.GetString(fila.GetOrdinal("Telefono")),
                    FechaRegistro =
fila.GetDateTime(fila.GetOrdinal("fechaRegistro")).ToString("yyyy/MM/dd
"),
                    esInactivo =
fila.GetBoolean(fila.GetOrdinal("esInactivo")),
                    esVip = fila.GetBoolean(fila.GetOrdinal("esVip")),
                };
            }
            return p;
        }


        public static int FindPorcentajeVip(string rut)
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand(@"SELECT * From ProveedorVip
WHERE idProveedor = @rut");
            cmd.Connection = cn;
            cmd.Parameters.AddWithValue("@rut", rut);
            try
            {
                cn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                {
                    if (dr.Read())
                    {
                        int porcentajeExtra;
                        {
                            porcentajeExtra =
```

```
Convert.ToInt32(dr["porcentajeExtra"]);
                        };
                        return porcentajeExtra;
                    }
                }
                return 0;
            }
            catch (Exception ex)
            {


                throw new Exception("No existe el Proveedor");


            }
            finally { cn.Close(); cn.Dispose(); }
        }
        #endregion
    }
}


**********************************
Filename: Servicio.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Diagnostics;


namespace Dominio
```

```
{
   public class Servicio : IActiveRecord
   {
       public string Nombre { get; set; }
       public string Foto { get; set; }
       public string Descripcion { get; set; }
       public List<TipoEvento> ListaTipoEventos = new
List<TipoEvento>();


       public override string ToString()
       {
           string ret = string.Format("{0}", Nombre);
           return ret;
       }


       #region Acceso a datos
       public bool Insertar()
       {
           throw new NotImplementedException();
       }


       public bool Eliminar()
       {
           throw new NotImplementedException();
       }


       public bool Modificar()
       {
           throw new NotImplementedException();
       }
       #endregion
```

```
        #region Finders

        public static Servicio FindByNombre(string nombre)
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand(@"SELECT * From Servicio
WHERE Nombre = @nombre");
            cmd.Connection = cn;
            cmd.Parameters.AddWithValue("@nombre", nombre);
            try
            {
                cn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                    if (dr.Read())
                    {
                        string nombreServicio =
dr.IsDBNull(dr.GetOrdinal("nombre")) ? "" :
dr.GetString(dr.GetOrdinal("nombre"));
                        string desc =
dr.IsDBNull(dr.GetOrdinal("Descripcion")) ? "" :
dr.GetString(dr.GetOrdinal("Descripcion"));
                        string foto =
dr.IsDBNull(dr.GetOrdinal("imagen")) ? "" :
dr.GetString(dr.GetOrdinal("imagen"));


                        Servicio s = new Servicio
                        {
                            Nombre = nombreServicio,
                            Descripcion = desc,
                            Foto = foto,
                            ListaTipoEventos = new List<TipoEvento>()
                        };
                        return s;
                    }
                return null;
```

```
            }
            catch (Exception ex)
            {
                throw new Exception("No existe el Servicio");
            }
            finally { cn.Close(); cn.Dispose(); }
        }


        public static List<Servicio> FindAll()
        {
            SqlConnection cn = Conexion.CrearConexion();
            SqlCommand cmd = new SqlCommand();


            cmd.CommandText = @"SELECT s.nombre AS Servicio,
s.descripcion AS 'Descripción del servicio', s.imagen as 'Foto',
t.nombre as 'Tipo de evento'
                                FROM Servicio AS s
                                INNER JOIN TipoEventoYServicio AS e ON
s.idServicio = e.idServicio
                                INNER JOIN TipoEvento AS t ON
e.idTipoEvento = t.idTipoEvento";
            cmd.Connection = cn;
            List<Servicio> listaServicios = null;
            try
            {
                Conexion.AbrirConexion(cn);
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                {
                    listaServicios = new List<Servicio>();
                    while (dr.Read())
                    {
                        Servicio s = CargarDatosDesdeReader(dr);
                        listaServicios.Add(s);
                    }
```

```
            }
            return listaServicios;
        }
        catch (SqlException ex)
        {
            //
            System.Diagnostics.Debug.Assert(false, ex.Message);
            return null;
        }
        finally
        {
            Conexion.CerrarConexion(cn);
        }
    }


    public static List<TipoEvento> FindTiposEventoByServicio(string
servicio)
    {
        SqlConnection cn = Conexion.CrearConexion();
        SqlCommand cmd = new SqlCommand();


        cmd.CommandText = @"SELECT t.nombre, t.descripción
                            FROM Servicio AS s
                            INNER JOIN TipoEventoYServicio AS e ON
s.idServicio = e.idServicio
                            INNER JOIN TipoEvento AS t ON
e.idTipoEvento = t.idTipoEvento
                            WHERE s.nombre = @servicio";


        cmd.Connection = cn;
        cmd.Parameters.AddWithValue("@servicio", servicio);
        List<TipoEvento> listaTipoEvento = null;
        try
        {
```

```
                Conexion.AbrirConexion(cn);
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.HasRows)
                {
                    listaTipoEvento = new List<TipoEvento>();
                    while (dr.Read())
                    {
                        Servicio s = Servicio.FindByNombre(servicio);
                        string tipo =
dr.IsDBNull(dr.GetOrdinal("nombre")) ? "" :
dr.GetString(dr.GetOrdinal("nombre"));
                        string desc =
dr.IsDBNull(dr.GetOrdinal("descripción")) ? "" :
dr.GetString(dr.GetOrdinal("descripción"));
                        TipoEvento t = new TipoEvento(tipo, desc);
                        listaTipoEvento.Add(t);
                    }
                }
                return listaTipoEvento;
            }
            catch (SqlException ex)
            {
                //
                System.Diagnostics.Debug.Assert(false, ex.Message);
                return null;
            }
            finally
            {
                Conexion.CerrarConexion(cn);
            }
        }


        protected static Servicio CargarDatosDesdeReader(IDataRecord
fila)
        {
            Servicio s = null;
```

```
            string nombreServicio =
fila.IsDBNull(fila.GetOrdinal("Servicio")) ? "" :
fila.GetString(fila.GetOrdinal("Servicio"));
            string desc = fila.IsDBNull(fila.GetOrdinal("Descripción del
servicio")) ? "" : fila.GetString(fila.GetOrdinal("Descripción del
servicio"));
            string foto = fila.IsDBNull(fila.GetOrdinal("Foto")) ? "" :
fila.GetString(fila.GetOrdinal("Foto"));


            if (fila != null)
            {
                s = new Servicio()
                {
                    Nombre = nombreServicio,
                    Descripcion = desc,
                    Foto = foto,
                    ListaTipoEventos =
FindTiposEventoByServicio(nombreServicio)
                };
            }
            return s;
        }
        #endregion


    }
}


**********************************
Filename: TipoEvento.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Diagnostics;


namespace Dominio
{
   public class TipoEvento
   {
       private string tipo;
       private string desc;


       public TipoEvento(string tipo, string desc)
       {
           this.tipo = tipo;
           this.desc = desc;
       }


       public string Nombre { get; set; }
       public string Descripcion { get; set; }


   }
}


**********************************
Filename: Usuario.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
```

```
using System.Threading.Tasks;


namespace Dominio
{
   public class Usuario
   {
       public string User { get; set; }
       public string Passw { get; set; }


       #region Encriptar Pass
       public static string EncriptarPassSHA512(string inputString)
       {
           SHA512 sha512 = SHA512.Create();
           byte[] bytes = Encoding.UTF8.GetBytes(inputString);
           byte[] hash = sha512.ComputeHash(bytes);
           return GetStringFromHash(hash);
       }


       private static string GetStringFromHash(byte[] hash)
       {
           StringBuilder result = new StringBuilder();
           for (int i = 0; i < hash.Length; i++)
           {
               result.Append(hash[i].ToString("X2"));
           }
           return result.ToString();
       }
       #endregion
   }
}
```

****

```
Filename: IInsertarProveedor.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;


namespace InsertarProveedor
{
   // NOTA: puede usar el comando "Rename" del menú "Refactorizar" para
cambiar el nombre de interfaz "IService1" en el código y en el archivo
de configuración a la vez.
   [ServiceContract]
   public interface IInsertarProveedor
   {


       [OperationContract]
       string GetData(int value);


       [OperationContract]
       CompositeType GetDataUsingDataContract(CompositeType composite);


       // TODO: agregue aquí sus operaciones de servicio
   }




   // Utilice un contrato de datos, como se ilustra en el ejemplo
siguiente, para agregar tipos compuestos a las operaciones de servicio.
   [DataContract]
```

```
    public class CompositeType
    {
        bool boolValue = true;
        string stringValue = "Hello ";


        [DataMember]
        public bool BoolValue
        {
            get { return boolValue; }
            set { boolValue = value; }
        }


        [DataMember]
        public string StringValue
        {
            get { return stringValue; }
            set { stringValue = value; }
        }
    }
}


**********************************
Filename: InsertarProveedor.svc.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
```

```
namespace InsertarProveedor
{
   // NOTA: puede usar el comando "Rename" del menú "Refactorizar" para
cambiar el nombre de clase "Service1" en el código, en svc y en el
archivo de configuración.
   // NOTE: para iniciar el Cliente de prueba WCF para probar este
servicio, seleccione Service1.svc o Service1.svc.cs en el Explorador de
soluciones e inicie la depuración.
   public class Service1 : IService1
   {
       public string GetData(int value)
       {
           return string.Format("You entered: {0}", value);
       }

       public CompositeType GetDataUsingDataContract(CompositeType
composite)
       {
           if (composite == null)
           {
               throw new ArgumentNullException("composite");
           }
           if (composite.BoolValue)
           {
               composite.StringValue += "Suffix";
           }
           return composite;
       }
   }
}
```

```
**********************************
Filename: IAgregarProv.cs
**********************************
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfAgregarProv
{
   [ServiceContract]
   public interface IAgregarProv
   {
       [OperationContract]
       bool InsertarProveedor(string rut, string nombreFantasia, string
email, string tel, double arancel, string fechaRegistro, bool
esInactivo, bool esVip, string pass);
   }
}


**********************************
Filename: ServicioAgregarProv.svc.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfAgregarProv
```

```
{
   // NOTA: puede usar el comando "Rename" del menú
"Refactorizar" para cambiar el nombre de clase "Service1" en el
código, en svc y en el archivo de configuración.
   // NOTE: para iniciar el Cliente de prueba WCF para probar
este servicio, seleccione Service1.svc o Service1.svc.cs en el
Explorador de soluciones e inicie la depuración.
   public class ServicioAgregarProv : IAgregarProv
   {
       public bool InsertarProveedor(string rut, string
nombreFantasia, string email, string tel, double arancel, string
fechaRegistro, bool esInactivo, bool esVip, string pass)
       {
           // Construyo un proveedor con los parámetros que
llegan desde el servicio
           Proveedor p = new Proveedor()
           {
               RUT = rut,
               NombreFantasia = nombreFantasia,
               Email = email,
               Telefono = tel,
               Arancelll = arancel,
               FechaRegistro = fechaRegistro,
               esInactivo = esInactivo,
               esVip = esVip
           };


           // Encripto el password y construyo un usuario
           string passEncriptada =
Usuario.EncriptarPassSHA512(pass);
           Usuario usu = new Usuario { User = rut, Passw =
passEncriptada };


           // Agrego el usuario al proveedor p
```

```
            p.AgregarUsuario(usu);


            return p.Insertar();
        }
    }
}


**********************************
Filename: IServicioListaProv.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfListaProv
{
    [ServiceContract]
    public interface IServicioListaProv
    {
        [OperationContract]
        IEnumerable<DtoProveedor> ObtenerProveedores();
    }


    [DataContract]
    public class DtoProveedor
    {
        [DataMember]
```

```
        public string RUT { get; set; }


        [DataMember]
        public string NombreFantasia { get; set; }


    }
}
```

```
**********************************
Filename: ServicioListaProv.svc.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfListaProv
{
    public class ServicioListaProv : IServicioListaProv
    {
        public IEnumerable<DtoProveedor> ObtenerProveedores()
        {
            List<Proveedor> listaCompleta = Proveedor.FindAll();
            if (listaCompleta == null) return null;
            List<DtoProveedor> proveedores = new
List<DtoProveedor>();
```

```
            foreach (Proveedor p in listaCompleta)
            {
                proveedores.Add(
                    new DtoProveedor()
                    {
                        NombreFantasia = p.NombreFantasia,
                        RUT = p.RUT
                    }
                );
            }
            return proveedores;
        }
    }
}


**********************************
Filename: IServicioListaProv.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfProveedores
{
    [ServiceContract]
    public interface IServicioListaProv
    {
        [OperationContract]
```

```
        IEnumerable<DtoProveedor> ObtenerProveedores();
    }


    [DataContract]
    public class DtoProveedor
    {
        [DataMember]
        public string RUT { get; set; }


        [DataMember]
        public string NombreFantasia { get; set; }


    }
}


**********************************
Filename: ServicioListaProv.svc.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfProveedores
{
    public class ServicioListaProv : IServicioListaProv
    {
```

```
    }
}


**********************************
Filename: IServicioCatalogoServicios.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfServicioCatalogoServicios
{
    // NOTA: puede usar el comando "Rename" del menú
"Refactorizar" para cambiar el nombre de interfaz "IService1" en
el código y en el archivo de configuración a la vez.
    [ServiceContract]
    public interface IServicioCatalogoServicios
    {
        [OperationContract]
        IEnumerable<DtoServicio> ObtenerServicios();
    }


    [DataContract]
    public class DtoServicio
    {
        [DataMember]
```

```
        public string Servicio { get; set; }


        [DataMember]
        public string Descripcion { get; set; }


        [DataMember]
        public string Foto { get; set; }


        [DataMember]
        public List<TipoEvento> TipoEvento { get; set; }
    }
}


**********************************
Filename: ServicioCatalogoServicios.svc.cs
**********************************
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Dominio;


namespace WcfServicioCatalogoServicios
{
    // NOTA: puede usar el comando "Rename" del menú
"Refactorizar" para cambiar el nombre de clase "Service1" en el
código, en svc y en el archivo de configuración.
    // NOTE: para iniciar el Cliente de prueba WCF para probar
```

```
este servicio, seleccione Service1.svc o Service1.svc.cs en el
Explorador de soluciones e inicie la depuración.
   public class ServicioCatalogoServicios :
IServicioCatalogoServicios
   {
       public IEnumerable<DtoServicio> ObtenerServicios()
       {
           List<Servicio> listaCompleta = Servicio.FindAll();
           if (listaCompleta == null) return null;
           List<DtoServicio> servicios = new List<DtoServicio>();
           foreach (Servicio s in listaCompleta)
           {
               List<TipoEvento> listaTipoEvento =
Servicio.FindTiposEventoByServicio(s.Nombre);
               servicios.Add(
                   new DtoServicio()
                   {
                       Servicio = s.Nombre,
                       Descripcion = s.Descripcion,
                       Foto = s.Foto,
                       TipoEvento = listaTipoEvento
                   }
               );
           }
           return servicios;
       }
   }
}
```