**CLASE CONEXIÓN**

**AGREGA CONEXION**

public static SqlConnection agregaConexion()

{

SqlConnection con;

try

{

con = new SqlConnection("Data Source = localhost; Initial Catalog = datosCursos; User ID = sa; Password = sqladmin");

con.Open();

MessageBox.Show("Si pude conectarme");

}

catch (Exception e)

{

con = null;

MessageBox.Show("No se pudo");

}

return con;

}

**LLENA COMBOBOX**

public static void llenarCombo(ComboBox cb)

{

SqlConnection con;

SqlDataReader rd;

try

{

con = Conexion.agregaConexion();

SqlCommand cmd = new SqlCommand("select nombre from estado", con);

rd = cmd.ExecuteReader();

while (rd.Read())

{

cb.Items.Add(rd.GetString(0));

}

cb.SelectedIndex = 0;

rd.Close();

con.Close();

}

catch (Exception e)

{

MessageBox.Show("No se pudo llenar el combo" + e);

}

}

**ALTA EN CLASE**

public String Alta(Curso c)

{

String res;

try

{

SqlConnection con;

con = Conexion.agregaConexion();

String queryMax = "select top(1) idCurso from curso order by idCurso desc";

SqlCommand cmd = new SqlCommand(queryMax, con);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.HasRows)

{

rd.Read();

int maximo = rd.GetInt32(0) + 1;

rd.Close();

String query2 = string.Format("insert into curso values " +

"({0}, '{1}', {2}, {3}) ", maximo, c.nombre, c.hora, c.idEstado);

SqlCommand cmd2 = new SqlCommand(query2, con);

cmd2.ExecuteNonQuery();

res = "Alta exitosa";

}

else

{

String query = string.Format("insert into curso values " +

"({0}, '{1}', {2}, {3}) ", 1, c.nombre, c.hora, c.idEstado);

SqlCommand cmd2 = new SqlCommand(query, con);

cmd2.ExecuteNonQuery();

res = "Alta exitosa";

}

con.Close();

}

catch (Exception e)

{

res = "Alta NO exitosa" + e.Message;

}

return res;

}

**ALTA EN BOTTON DE VENTANA**

private void BtAgregar\_Click(object sender, RoutedEventArgs e)

{

try

{

Curso c = new Curso(txNombre.Text, Int32.Parse(txHoras.Text), cbEstados.SelectedIndex);

MessageBox.Show(c.Alta(c));

}

catch(Exception ex)

{

MessageBox.Show("No se pudo Agregar" + ex);

}

}

**BAJA EN CLASE**

public String darDeBaja(String nombre)

{

String res = "";

try

{

SqlConnection con = Conexion.agregaConexion();

String query = String.Format("delete from aspirante where nombre='{0}'", nombre);

SqlCommand cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

con.Close();

res = "Baja exitosa";

}

catch (Exception e)

{

res = "No se pudo dar de baja" + e.ToString();

}

return res;

}

**BAJA EN BOTTON**

private void btBaja\_Click(object sender, RoutedEventArgs e)

{

try

{

Aspirantes a = new Aspirantes();

a.darDeBaja(cbClaveUica.SelectedItem.ToString());

}catch(Exception ex)

{

MessageBox.Show("No se pudo Eliminar" + ex);

}

}

**GENERAR REPORTE EN CLASE**

public String generarReporte(int idEdo, DataGrid dg)

{

String res = " ";

try

{

SqlConnection con = Conexion.agregaConexion();

String query = String.Format("select \* from curso where idEdo={0}",idEdo);

SqlCommand com = new SqlCommand(query, con);

SqlDataReader dr = com.ExecuteReader();

dg.ItemsSource = dr;

res = "Éxito";

}

catch (Exception e)

{

res = "Falló";

}

return res;

}

**REPORTE EN BOTTON DE VENTANA**

private void BtReporte\_Click(object sender, RoutedEventArgs e)

{

try

{

Curso c;

c = new Curso();

c.generarReporte(cbEstados.SelectedIndex, dgCursos);

}catch(Exception ex)

{

MessageBox.Show("No se puede mostrar reporte " + ex);

}

}

**BOTTON REGRESAR**

private void BtRegresar\_Click(object sender, RoutedEventArgs e)

{

MainWindow main = new MainWindow();

main.Show();

this.Hide();

}

**BOTTON CAMBIAR VENTANA**

private void BtModificar\_Click(object sender, RoutedEventArgs e)

{

Modificar m = new Modificar();

m.Show();

this.Hide();

}

**WEB**

//Conexión

protected OdbcConnection conectarBD()

{

String stringConexion = "Driver={SQL Server Native Client 11.0};Server=112SALAS10;Uid=sa;Pwd=sqladmin;Database=GameSpot";

try

{

OdbcConnection conexion = new OdbcConnection(stringConexion);

conexion.Open();

lbContador.Text = "conexion exitosa";

return conexion;

}

catch (Exception ex)

{

lbContador.Text = ex.StackTrace.ToString();

return null;

}

//Page Load

protected void Page\_Load(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

}

//Boton para checar contraseña y mandar a la siguiente pagina

protected void btPagina2\_Click(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

String query = " select claveU from usuario where email = '" + txUsuario.Text + "'and password = '" + txContraseña.Text + "'";

OdbcCommand cmd = new OdbcCommand(query, miConexion);

OdbcDataReader rd = cmd.ExecuteReader();

if (rd.HasRows)

{

rd.Read();

Session["claveUnica"] = rd.GetInt32(0).ToString(); //se guarda para otras paginas

Response.Redirect("Pagina2.aspx"); //Manda a otra pagina

}

else

lbContador.Text = " el usuario o contraseña son incorrectos";

rd.Close();

}

}

//Page Load que llenas textBoxes

protected void Page\_Load(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

String query = String.Format("select nombre,edad,sexo from usuario where claveU ='{0}'", Session["claveUnica"].ToString());

OdbcCommand cmd = new OdbcCommand(query, miConexion);

OdbcDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

txNombre.Text = rd.GetString(0);

txEdad.Text = rd.GetString(1);

txSexo.Text = rd.GetString(2);

rd.Close();

}

else

lbContador.Text = " error";

}

else

lbContador.Text = " no se pudo conectar ";

}

//LLena DropDownList en el Page\_Load - AutoPostBack True

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack) {

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

String query = String.Format("select juegos.nombre from juegos,escriben where escriben.claveU = {0} and escriben.claveJ = juegos.claveJ", Session["claveUnica"].ToString());

OdbcCommand cmd = new OdbcCommand(query, miConexion);

OdbcDataReader rd = cmd.ExecuteReader();

ddJuegos.Items.Clear();

while (rd.Read())

{

ddJuegos.Items.Add(rd.GetString(0));

}

rd.Close();

}

}

}

//cambiar label con querys

protected void ddJuegos\_SelectedIndexChanged(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

String query = String.Format("select claveJ from juegos where nombre= '{0}'", ddJuegos.SelectedValue);

OdbcCommand cmd = new OdbcCommand(query, miConexion);

OdbcDataReader rd = cmd.ExecuteReader();

rd.Read();

int claveJuego = rd.GetInt16(0);

String query2 = String.Format("select critica.contentido from critica where critica.claveC =(select claveC from escriben where escriben.claveu= {0} and escriben.claveJ={1})", Session["claveUnica"].ToString(), claveJuego);

OdbcCommand cmd2 = new OdbcCommand(query2, miConexion);

OdbcDataReader rd2 = cmd2.ExecuteReader();

rd2.Read();

lbCritica.Text = rd2.GetString(0);

rd.Close();

rd2.Close();

}

}

// llenar GridView

protected void ddJuegos\_SelectedIndexChanged(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

String query = String.Format("select claveJ from juegos where nombre= '{0}'", ddJuegos.SelectedValue);

OdbcCommand cmd = new OdbcCommand(query, miConexion);

OdbcDataReader rd = cmd.ExecuteReader();

rd.Read();

int claveJuego = rd.GetInt16(0);

MessageBox.Show(claveJuego.ToString());

String query2 = String.Format("select nombre, resumen, consola, fechaLanzamiento from juegos where claveJ ={0}", claveJuego);

OdbcCommand cmd2 = new OdbcCommand(query2, miConexion);

OdbcDataReader rd2 = cmd2.ExecuteReader();

gvJuegos.DataSource = rd2;

gvJuegos.DataBind();

rd.Close();

rd2.Close();

}

}

**HTML**

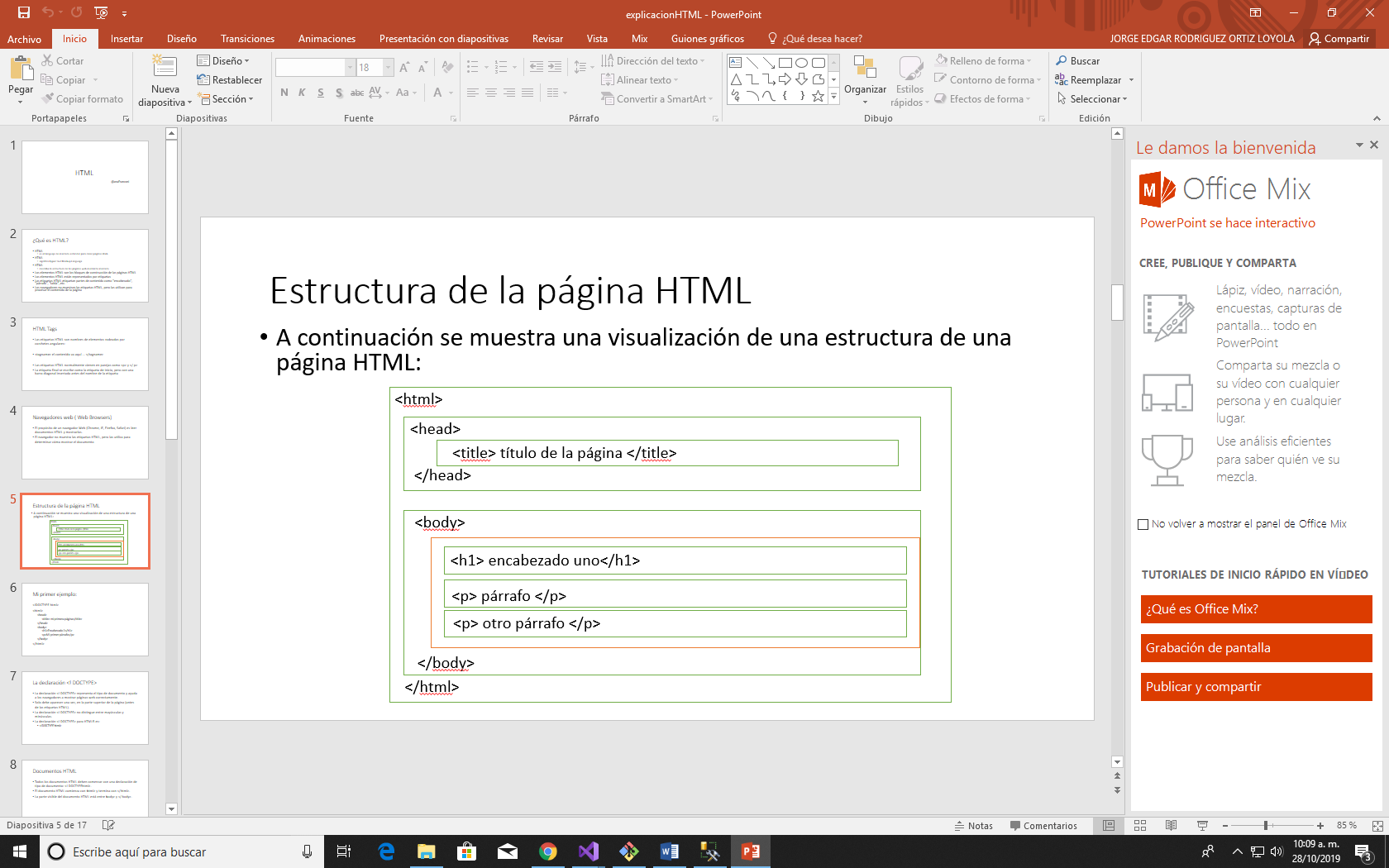
* HTML
  + es el lenguaje de marcado estándar para crear páginas Web
* HTML
  + significa Hyper Text Markup Language
* HTML
  + describe la estructura de las páginas web mediante marcado
* Los elementos HTML son los bloques de construcción de las páginas HTML
* Los elementos HTML están representados por etiquetas
* Las etiquetas HTML etiquetan partes de contenido como "encabezado", "párrafo", "tabla", etc.
* Los navegadores no muestran las etiquetas HTML, pero las utilizan para procesar el contenido de la página

TAGS

* Las etiquetas HTML son nombres de elementos rodeados por corchetes angulares:
* <tagname> el contenido va aquí ... </ tagname>
* Las etiquetas HTML normalmente vienen en parejas como <p> y </ p>
* La etiqueta final se escribe como la etiqueta de inicio, pero con una barra diagonal insertada antes del nombre de la etiqueta

EXPLORADORES WEB

* El propósito de un navegador Web (Chrome, IE, Firefox, Safari) es leer documentos HTML y mostrarlos.
* El navegador no muestra las etiquetas HTML, pero las utiliza para determinar cómo mostrar el documento



<! DOCTYPE>

* La declaración <! DOCTYPE> representa el tipo de documento y ayuda a los navegadores a mostrar páginas web correctamente.
* Solo debe aparecer una vez, en la parte superior de la página (antes de las etiquetas HTML).
* La declaración <! DOCTYPE> no distingue entre mayúsculas y minúsculas.
* La declaración <! DOCTYPE> para HTML5 es:
  + <!DOCTYPE html>