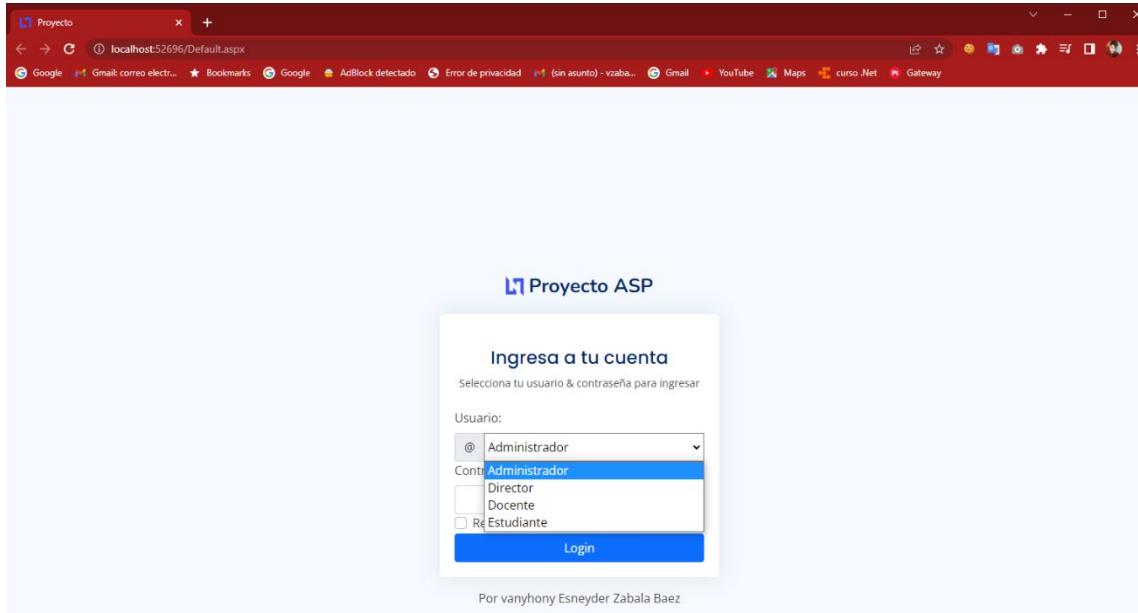
The background of the page features a large, abstract graphic composed of several overlapping, translucent blue and white wavy lines. These lines create a sense of depth and motion, resembling flowing water or light. They are primarily located on the left side of the page, with some faint lines extending towards the right.

Manual de Usuario Proyecto  
ASP.NET Web Forms

# Proyecto ASP

Vanyhony Esneyder Zabala Baez

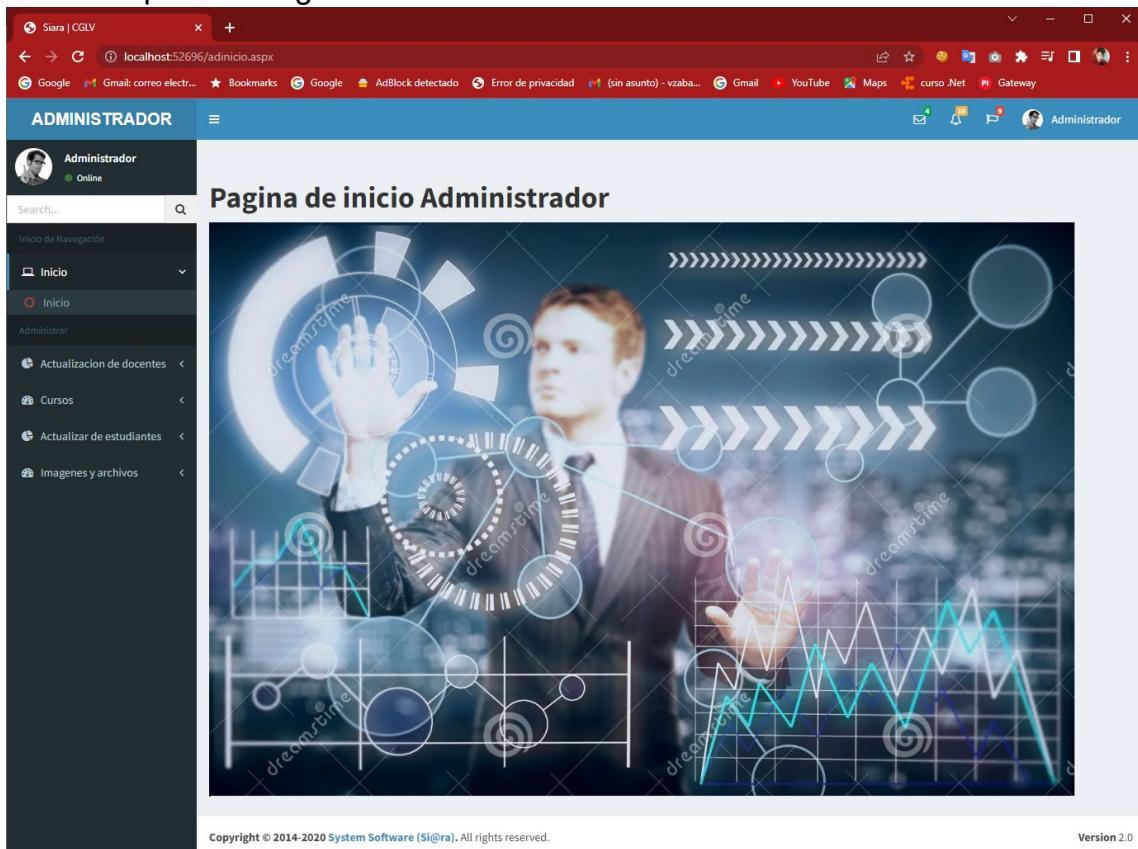
Al iniciar la página se visualizará el Login para ingresar al sistema, en el usuario se encontrará 4 usuarios, Administrador, Director, Docente y Estudiante



### Administrador:

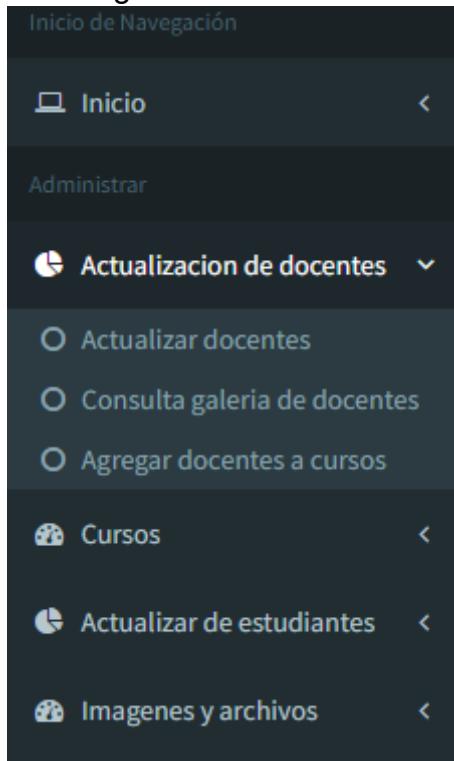
Para el administrador solo le damos Login.

Ya estando adentro de la plantilla de administrador, nos mostrara la pagina de inicio que es la siguiente:



En el menú se encontrará los ítems disponibles para el administrador como se evidencia en la imagen anterior.

Para la administración de docentes, se desplegará Actualización de docentes y se mostrarán los siguientes ítems:



Al seleccionar Actualizar docentes, se mostrará la información de los docentes, divididas por paginación, y cada uno con sus botones de Editar, Eliminar y Nuevo por si se quiere crear un nuevo docente:

A screenshot of a web application window titled 'Sara | CGUV'. The address bar shows 'localhost:52656/Actualizadocentes.aspx'. The main content area has a blue header 'ADMINISTRADOR'. On the left is a sidebar with a user profile picture and the name 'Administrador'. Below the sidebar are several menu items: 'Inicio', 'Actualizar de docentes' (selected), 'Cursos', 'Actualizar de estudiantes', and 'Imagenes y archivos'. The main content area displays a table with teacher information. The table has columns for Código Usuario, Código Docente, Apellido 1, Apellido 2, Imagen, and Archivo. The data shown is: Código Usuario: 3, Código Docente: 52322195, Apellido 1: Aceró, Apellido 2: Rivera, Imagen: dos.png, Archivo: informe2.pdf. Below the table are three buttons: 'Editar' (red), 'Eliminar' (blue), and 'Nuevo' (green). A status bar at the bottom of the table area says '12345678910...'.

En la parte superior de la tabla que se muestra los docentes, se encuentra la opción de buscar un docente por código.

Se escribirá el código de un docente luego darle en buscar y el sistema hará una búsqueda de los docentes hasta encontrar el docente con este

código y mostrarlo en pantalla, solo con las opciones de Editar y eliminar y un botón para volver a la pestaña anterior.

The screenshot shows a web-based administration interface titled 'ADMINISTRADOR'. On the left is a sidebar with a user profile picture and the name 'Administrador' (status: Online). Below it are navigation links: 'Inicio', 'Administrar', 'Actualización de docentes', 'Cursos', 'Actualizar de estudiantes', and 'Imagenes y archivos'. The main content area has a header 'Buscar Docente por Código:' with a search input field containing '52322195', a green 'Buscar' button, and a red 'Volver' button. Below this is a table titled 'Actualización de docentes' with the following data:

Código Usuario:	3	Apellido 1:	Acero	Imagen:
Código Docente:	52322195	Apellido 2:	Rivera	Archivo:
Teléfono:	2	Nombre 1:	Gladys	
Ingreso:	78276683	Nombre 2:	Elena	

At the bottom of the table are two buttons: 'Editar' (highlighted in blue) and 'Eliminar'.

En caso de no encontrar ningún docente se mostrará una alerta que dirá que no se encontró docente con dicho código y al darle aceptar retornara la pagina de todos los docentes:

A screenshot of a browser window titled 'Siara | CGLV'. The address bar shows 'localhost:52696/Adocentes.aspx'. The page content displays an alert dialog box from 'localhost:52696' with the message 'No se encontro docente con este codigo' (No teacher found with this code) and a blue 'Aceptar' (Accept) button. The browser toolbar at the top includes icons for Google, Gmail, Bookmarks, AdBlock detected, and Error de privacidad.

### Explicación técnica de este evento:

En el evento del botón buscar empecé creando una condicional donde me validara que el campo no este vacío y si lo esta me retorno la misma pagina donde estoy, si esta llena, va abrir la conexión a base de datos y empezara a buscar en la tabla de docentes, un registro con el mismo código, si no lo encuentra, genera un Script de alerta mostrando el error de que no se encontró, si lo encuentra, guardara la variable de código en una sesión, y redirigirá a otra plantilla que se creó:

```
protected void Button2_Click(object sender, EventArgs e)
{
    conexion.Open();
    if (TextBox4.Text.Equals(""))
    {
        Response.Redirect("Acdocentes.aspx");
    }
    else
    {
        string cadena = "select * from tbldocentes";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        SqlDataReader registros = comando.ExecuteReader();
        while (registros.Read())
        {
            if (registros["doccodigo"].ToString().Equals(TextBox4.Text))
            {
                codigoDocente = (registros["doccodigo"].ToString());
                condicion = true;
            }
        }
        if (condicion == false)
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(),
                "alert",
                "alert('No se encontro docente con este codigo');", true);
            conexion.Close();
            TextBox4.Text = "";
        }
        else
        {
            //Almacenamos las dos variables de sesion
            Session["codigoDocente"] = codigoDocente;
            //Redireccionamos a la siguiente pagina

            Response.Redirect("DocinicioConsulta.aspx");

            conexion.Close();
        }
    }
}
```

En la otra plantilla que cree, agregue un Form View apuntando a la tabla de docentes, agregando el “**Where**” a la sentencia Sql refiriéndose a la variable de sesión del código del docente.

Al cargar la plantilla del docente consultado, revisara que halla mas de una variable de sesión si no retornara la plantilla anterior, esto para evitar errores, si existe una Asignara la variable de sesión al valor del textBox para que se pueda seguir haciendo consultas en esta plantilla, y una condición donde si se ingresa otro valor al text box lo asigne a la variable de sesión para seguir buscando

```

protected void Page_Load(object sender, EventArgs e)
{
    if (this.Session.Count == 0)
    {
        Response.Redirect("Acdocentes.aspx");
    }
    else
    {
        if (TextBox4.Text.Equals("")) {
            TextBox4.Text = this.Session["codigoDocente"].ToString();
        }
        else
        {
            this.Session["codigoDocente"] = TextBox4.Text;
        }
    }
}

```

### Crear docentes:

Al seleccionar el botón se abrirá el formulario de creación de un nuevo docente.

Si ingresa todos los datos sin foto y ningún archivo, al darle insertar, el sistema validara que no exista otro docente con el mismo código y lo insertara y retornara a la pantalla anterior con un mensaje de que se creó correctamente:

**Crear Docente:**

Código Usuario:	Docente	Apellido 1:	Zabala	Imagen:	Seleccionar archivo
Código Docente:	114	Apellido 2:	Baez	Archivo:	Seleccionar archivo
Teléfono:	3203590398	Nombre 1:	Vanyhony		
Ingreso:	1516	Nombre 2:	Esneyder		

**Actualización de Docentes:**

Código Usuario:	3	Apellido 1:	Aceró	Imagen:	uno.png
Código Docente:	49389877	Apellido 2:	Rivera	Archivo:	informe1.pdf
Teléfono:	1	Nombre 1:	Carlos		
Ingreso:	56987736	Nombre 2:	Arturo		

Success Message: Se a guardado con éxito el registro.

Si encuentra un docente que tenga el mismo código que se está ingresando nos mostrara el error y no dejara ingresar el nuevo docente hasta cambiar el código.

The screenshot shows a 'Crear Docente' (Create Teacher) form. The fields filled are: Código Usuario: Docente, Código Docente: 49389877, Telefono: 3203590398, Ingreso: 1516. The error message 'Error, el codigo de este docente ya existe' (Error, this teacher's code already exists) is displayed at the bottom.

Al insertar una imagen o un archivo o los dos, el sistema hará ciertas validaciones, donde verificará la extensión de los archivos subidos, y su peso (debe ser máximo de 200kb), si una de estas validaciones no cumple, mostrará error y no dejará insertar el registro.

The screenshot shows the same 'Crear Docente' form. The error message 'La extencion de la imagen no es valida' (The image extension is not valid) is shown above another message 'La extencion del archivo no es valida, debe ser un archivo .pdf' (The file extension is not valid, it must be a .pdf file). Both messages are in red.

En algunos casos pueda que el nombre de la imagen o del archivo, sea el mismo a uno que exista en el sistema, en tal caso, el nombre de la imagen le agregara un – y un numero consecutivo para que así no se repita, se evidencia al momento de consultar al docente que se ingresó.

The screenshot shows the 'Actualización de docentes' (Teacher Update) page. The right side has two input fields: 'Imagen:' with value 'uno-1.png' and 'Archivo:' with value 'informe3-2.pdf'. These fields are highlighted with a blue border.

### Explicación técnica para crear nuevos registros.

Al usuario darle Insertar entrara a un evento donde capturara todos los datos ingresados por el usuario, luego en una condicional validara el código del docente , dirigiéndolo a otro método para validar que no se repetida , este método retornara un String donde si se repite retorna “repetidas” si no retornara una cadena vacía, se guarda el retorno en una variable y se

verificara, si no son repetidas, validara cuatro condiciones, la primera es si el usuario no inserto imagen ni archivo, en ese caso lo insertara a la base de datos el registro, la segunda si ingreso imagen pero no un archivo, aquí entrara a un método para subir imágenes donde se validara las condiciones de peso y extensión, y me retornara un String, existirá una variable para la clase del nombre de la imagen, dependiendo del retorno del método de imágenes compara si la variable y el retorno son iguales, si lo son dejara insertar los datos a la base de datos, en el método de imágenes, hará las validaciones si la imagen existe y si es así cambiara el nombre aumentando atreves de un contador hasta que ya no se repita , validando así por medio de un While. Para la tercera condición, es por si el usuario ingreso un archivo pero no una imagen, hará el mismo procedimiento que la condición anterior pero con un método para subir archivos, y la última condición es por si el usuario inserta una imagen y un archivo, hará los mismo procesos que los anteriores condicionales, solo que aquí validara si la imagen o el archivo esta correcto pero el otro no, eliminara el archivo del servidor ya que se inserta en los métodos, así para que no exista archivos en el servidor sin usar.

### Evento Insertar:

```
// Insertar
protected void Button3_Click(object sender, EventArgs e)
{
    string exception;
    DropDownList usuCodigo = FormView1.FindControl("usuCodigoDropDownList") as DropDownList;
    TextBox docCodigo = FormView1.FindControl("docCodigoTextBox") as TextBox;
    TextBox telefono = FormView1.FindControl("telefonoTextBox") as TextBox;
    TextBox docNombre1 = FormView1.FindControl("docNombre1TextBox") as TextBox;
    TextBox docApellido1 = FormView1.FindControl("docApellido1TextBox") as TextBox;
    TextBox docApellido2 = FormView1.FindControl("docApellido2TextBox") as TextBox;
    TextBox docNombre2 = FormView1.FindControl("docNombre2TextBox") as TextBox;
    FileUpload docImagen = FormView1.FindControl("docImagenFileUpload") as FileUpload;
    FileUpload docArchivo = FormView1.FindControl("docArchivoFileUpload1") as FileUpload;
    Label label8 = FormView1.FindControl("Label8") as Label;
    Label label5 = FormView1.FindControl("Label5") as Label;
    Label label6 = FormView1.FindControl("Label6") as Label;
    Label label7 = FormView1.FindControl("Label7") as Label;

    String imagenVar = "";
    String archivoVar = "";
    exception = validarLlavesRepetidas(docCodigo);
    if (exception.Equals("repetidas"))
    {
        label8.Text = "Error, el código de este docente ya existe";
    }
    else
    {
        if (docImagen.FileName.Equals("") && docArchivo.FileName.Equals(""))
        {
            label5.Text = "";
            label6.Text = "";
            label7.Text = "";
            conexion.Open();
            string cadena = "INSERT INTO tbldocentes (usuCodigo, docCodigo, docApellido1, docApellido2, docNombre1, docNombre2, docingreso, dotelefono) VALUES({int32.Parse(usuCodigo.Text)}, {int32.Parse(docCodigo.Text)},";
            Comando.ExecuteNonQuery();
            conexion.Close();
            Session["guardadoDoc"] = "guardado";
            Response.Redirect("Adocentes.aspx");
        }
        else if ((docImagen.FileName.Equals("") && docArchivo.FileName.Equals("")))
        {
            label5.Text = "";
            label6.Text = "";
            label7.Text = "";
            imagenVar = docImagen.FileName;
        }
    }
}
```

```

        if (label5.Text.Equals(nombreImagen))
        {
            conexion.Open();
            string cadena = $"INSERT INTO tbldocentes (usuCodigo, docCodigo, docApellido1, docApellido2, docNombre1, docNombre2, docIngreso, docTelefono, imagen) VALUES({Int32.Parse(usuCodigo.Text)}, {Int32.Parse(docCodigo.Text)}, {docApellido1}, {docApellido2}, {docNombre1}, {docNombre2}, {docIngreso}, {docTelefono}, {imagen});";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            comando.ExecuteNonQuery();
            comando.Close();
            Session["guardadoDoc"] = "guardado";
            Response.Redirect("Adocentes.aspx");
        }
    }
    else if (docImagen.FileName.Equals("") && !docArchivo.FileName.Equals(""))
    {
        label5.Text = "";
        label6.Text = "";
        label8.Text = "";
        archivoVar = docArchivo.FileName;
        label6.Text = subirArchivo(archivoVar, docArchivo);
        if (label6.Text.Equals(nombreArchivo))
        {
            conexion.Open();
            string cadena = $"INSERT INTO tbldocentes (usuCodigo, docCodigo, docApellido1, docApellido2, docNombre1, docNombre2, docIngreso, docTelefono, archivo) VALUES({Int32.Parse(usuCodigo.Text)}, {Int32.Parse(docCodigo.Text)}, {docApellido1}, {docApellido2}, {docNombre1}, {docNombre2}, {docIngreso}, {docTelefono}, {archivo});";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            comando.ExecuteNonQuery();
            comando.Close();
            Session["guardadoDoc"] = "guardado";
            Response.Redirect("Adocentes.aspx");
        }
    }
    else if (!docImagen.FileName.Equals("") && !docArchivo.FileName.Equals(""))
    {
        label5.Text = "";
        label6.Text = "";
        label8.Text = "";
        imagenVar = docImagen.FileName;
        archivoVar = docArchivo.FileName;
        label5.Text = subirImagen(imagenVar, docImagen);
        condicion2 = true;
        label6.Text = subirArchivo(archivoVar, docArchivo);
        if (label5.Text.Equals(nombreImagen) && label6.Text.Equals(nombreArchivo))
        {
            conexion.Open();
            string cadena = $"INSERT INTO tbldocentes (usuCodigo, docCodigo, docApellido1, docApellido2, docNombre1, docNombre2, docIngreso, docTelefono, imagen, archivo) VALUES({Int32.Parse(usuCodigo.Text)}, {Int32.Parse(docCodigo.Text)}, {docApellido1}, {docApellido2}, {docNombre1}, {docNombre2}, {docIngreso}, {docTelefono}, {imagen}, {archivo});";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            comando.ExecuteNonQuery();
            comando.Close();
            Session["guardadoDoc"] = "guardado";
            Response.Redirect("Adocentes.aspx");
        }
    }
    if(label5.Text.Equals(nombreImagen) && !label6.Text.Equals(nombreArchivo))
    {
        label5.Text = "";
        if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + nombreImagen))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + nombreImagen);
        }
        if (!label5.Text.Equals(nombreImagen) && label6.Text.Equals(nombreArchivo))
        {
            label6.Text = "";
            if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + nombreArchivo))
            {
                System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + nombreArchivo);
            }
        }
    }
}

```

## Método subir imagen:

```

4 referencias
public string subirImagen(string imagenVar, FileUpload docImagen)
{
    String label5="";
    extImg = System.IO.Path.GetExtension(imagenVar);
    if (extImg.Equals(".bmp") || extImg.Equals(".gif") || extImg.Equals(".tif") || extImg.Equals(".png"))
    {
        while (condicion2 == true)
        {
            if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + imagenVar))
            {
                string variableNombreSinExt = System.IO.Path.GetFileNameWithoutExtension(docImagen.FileName);
                contador++;
                imagenVar = $"{variableNombreSinExt}-{contador}{extImg}";
            }
            else
            {
                condicion2 = false;
            }
        }
        if (docImagen.PostedFile.ContentLength > 200000)
        {
            tamanoMb = ((double)docImagen.PostedFile.ContentLength / (double)1000) / 1000;
            label5 = $"Se supero el peso de subida para la imagen, el peso debe ser menor de 0.2mb (200kb), se subio con {tamanoMb.ToString()} MB";
            return label5;
        }
        else
        {
            docImagen.SaveAs(Server.MapPath(".") + "/imagenes/" + imagenVar); //guarda el archivo contenido en FileUpload1
            nombreImagen = imagenVar;
            return imagenVar;
        }
    }
    else
    {
        label5 = "La extencion de la imagen no es valida";
        return label5;
    }
}

```

## Método subir archivo:

```
4 referencias
public string subirArchivo(string archivoVar, FileUpload docArchivo)
{
    string label6 = "";
    extArc = System.IO.Path.GetExtension(archivoVar);
    if (extArc.Equals(".pdf"))
    {
        while (condicion2 == true)
        {
            if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + archivoVar))
            {
                string variableNombreSinExt = System.IO.Path.GetFileNameWithoutExtension(docArchivo.FileName);
                contador++;
                archivoVar = $"{variableNombreSinExt}-{contador}{extArc}";
            }
            else
            {
                condicion2 = false;
            }
        }
        if (docArchivo.PostedFile.ContentLength > 200000)
        {
            tamanoMb = ((double)docArchivo.PostedFile.ContentLength / (double)1000) / 1000;
            label6 = $"Se supero el peso de subida para el archivo, el peso debe ser menor de 0.2mb (200kb), se subio con {tamanoMb.ToString()} MB";
            return label6;
        }
        else
        {
            docArchivo.SaveAs(Server.MapPath(".") + "/archivos/" + archivoVar); //guarda el archivo contenido en FileUpload1
            nombreArchivo = archivoVar;
            return archivoVar;
        }
    }
    else
    {
        label6 = "La extencion del archivo no es valida, debe ser un archivo .pdf";
        return label6;
    }
}
```

## Método validar llaves:

```
1 referencia
public string validarLLavesRepetidas(TextBox docCodigo)
{
    conexion.Open();
    string cadena = "select doccodigo from tbldocentes";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    SqlDataReader registros = comando.ExecuteReader();
    string error = "";
    while (registros.Read())
    {
        if (registros["doccodigo"].ToString().Equals(docCodigo.Text))
        {
            error = "repetidas";
        }
    }
    conexion.Close();
    return error;
}
```

Cuando todo esta correcto, retorna a la pagina de todos los docentes, y se creo un Label que esta invisible, al todo estar correcto se creo una variable de Session que dependiendo lo que diga activara el Label al cargar la pagina y con un botón los deshabilita:

Al cargar la página:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["guardadoDoc"]!=null)
    {
        if (Session["guardadoDoc"].ToString().Equals("guardado")){
            Label7.Text = "Se a guardado con exito el registro";
            Label7.Visible = true;
            Button4.Visible = true;
            Session["guardadoDoc"] = "";
        }
    }
}
```

Botón para cerrar el mensaje:

```
0 referencias
protected void Button4_Click(object sender, EventArgs e)
{
    Label7.Visible = false;
    Button4.Visible = false;
}
```

### Eliminar docentes:

Si se desea eliminar un docente, asegurarse primero que no tenga ningún curso asignado de lo contrario el sistema no lo permitirá, si el docente no tiene ningún curso asignado solo es buscarlo ya sea por la paginación o por su código y seleccionar el botón eliminar, si se elimino se mostrara un mensaje que se eliminó correctamente.

Actualizar de docentes

Código Usuario:	3	Apellido 1:	Zabala	Imagen:	uno-1.png
Código Docente:	114	Apellido 2:	Baez	Archivo:	informe3-2.pdf
Teléfono:	3203590398	Nombre 1:	Vanyhony		
Ingreso:	1516	Nombre 2:	Esneyder		

Buscar Docente por Código:

Actualizar de Docentes

Código Usuario:	3	Apellido 1:	Aereo	Imagen:	uno.png
Código Docente:	49389877	Apellido 2:	Rivera	Archivo:	informe1.pdf
Teléfono:	1	Apellido 1:	Carlos		
Ingreso:	56987736	Apellido 2:	Arturo		

Buscar Docente por Código:

12345678910...

Se a eliminado con exito el registro

### Explicación técnica de eliminación de docentes:

Para el evento de eliminación primero se verifica si el usuario tenia alguna imagen o un archivo registrado, si es así, buscarlo en el servidor y eliminarlo, después, con la conexión abierta de la base de datos, se ejecuta la sentencia sql para eliminar el docente obteniendo el código, y la variable de sesión misma asignada para insertar, obtener el valor “eliminado” para que cuando cargue la página muestre el mensaje.

### Evento eliminar:

```
0 referencias
protected void DeleteButton_Click(object sender, EventArgs e)
{
    conexion.Open();
    Label docImagen = FormView1.FindControl("docImagenLabel") as Label;
    Label docArchivo = FormView1.FindControl("docArchivoLabel") as Label;
    if (!docImagen.Text.Equals(""))
    {
        //borrar imagen del directorio

        //Dado el caso, verifico que exista el archivo..

        if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + docImagen.Text))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + docImagen.Text);
        }
    }
    if (!docArchivo.Text.Equals(""))
    {
        //borrar archivo del directorio

        //Dado el caso, verifico que exista el archivo..

        if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + docArchivo.Text))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + docArchivo.Text);
        }
    }
    Label docCodigo = FormView1.FindControl("docCodigoLabel") as Label;
    string cadena = $"DELETE FROM tbldocentes WHERE doccodigo = {docCodigo.Text}";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    comando.ExecuteReader();
    Session["guardadoDoc"] = "eliminado";
    conexion.Close();
    Response.Redirect("Acdocentes.aspx");
}
```

Al cargar página:

```
else if (Session["guardadoDoc"].ToString().Equals("eliminado"))
{
    Label7.Text = "Se a eliminado con exito el registro";
    Label7.Visible = true;
    Button4.Visible = true;
    Session["guardadoDoc"] = "";
}
```

### Editar docente:

Para editar un docente, se busca por la paginación o por su código, luego seleccionar el botón editar, modificamos los campos que requiera actualizar, y darle actualizar, si todo es correcto retornara la pagina de todos los docentes y mostrara un mensaje que se actualizo correctamente

Buscar Docente por Codigo:  **Buscar**

**Actualizacion de Docentes**

Codigo Usuario:	<input type="text" value="Docente"/>	Apellido 1:	<input type="text" value="Bohorquez"/>
Codigo Docente:	<input type="text" value="84577705"/>	Apellido 2:	<input type="text" value="Rodriguez"/>
Telefono:	<input type="text" value="5"/>	Nombre 1:	<input type="text" value="Jeffer Editato"/>
Ingreso:	<input type="text" value="312455100"/>	Nombre 2:	<input type="text"/>

**Actualizar**
**Cancelar**

...45678910111213

**Actualizacion de Docentes**

Codigo Usuario:	3	Apellido 1:	Acero	Imagen:	<input type="text" value="uno.png"/>
Codigo Docente:	49389877	Apellido 2:	Rivera	Archivo:	informe1.pdf
Telefono:	1	Apellido 1:	Carlos		
Ingreso:	56987736	Nombre 2:	Arturo		

**Editar**
**Eliminar**
**Nuevo**

12345678910...

**Se a actualizado con exito el registro**

Al actualizar la imagen o un archivo hará las mismas validaciones que al crear un docente.

Si desea eliminar la imagen o el archivo de un docente darle al botón (eliminar imagen y actualizar) o si es el archivo el del archivo, estos botones solo actualizasen el campo de la imagen o archivo mas no los otros.

Buscar Docente por Codigo:  **Buscar** **Volver**

**Actualizacion de docentes**

Codigo Usuario:	3	Apellido 1:	Bohorquez	Imagen:
Codigo Docente:	84577705	Apellido 2:	Rodriguez	Archivo:
Telefono:	5	Nombre 1:	<input type="text" value="Jeffer Editato"/>	
Ingreso:	312455100	Nombre 2:		

**Editar**
**Eliminar**
**Nuevo**

12345678910...

**Actualización de Docentes**

Código Usuario:	Docente	Apellido 1:	Acer	Imagen:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Código Docente:	49389877	Apellido 2:	Rivera	Archivo:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Teléfono:	1	Nombre 1:	Carlos			
Ingreso:	56987736	Nombre 2:	Arturo			

**Actualizar** **Cancelar**

12345678910...

**Actualización de Docentes**

Código Usuario:	3	Apellido 1:	Acer	Imagen:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Código Docente:	49389877	Apellido 2:	Rivera	Archivo:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Teléfono:	1	Nombre 1:	Carlos			
Ingreso:	56987736	Nombre 2:	Arturo			

**Editar** **Eliminar** **Nuevo**

12345678910... X

Se ha eliminado la foto del registro con éxito

Si el docente no tiene ningún archivo o imagen registrado actualmente, y si se da a los botones de eliminación de archivo, mostrara error.

**Actualización de Docentes**

Código Usuario:	Docente	Apellido 1:	Bohorquez	Imagen:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Código Docente:	84577705	Apellido 2:	Rodriguez	Archivo:	<input type="button" value="Seleccionar archivo"/>	Sin arc...ionados
Teléfono:	5	Nombre 1:	Jeffer Editato			
Ingreso:	312455100	Nombre 2:				

**Actualizar** **Cancelar** **No existe ninguna foto insertada en este registro**

Y se puede insertar una o hacer la modificación necesaria y darle actualizar.

### Explicación técnica para actualizar un registro.

Para actualizar se implemento en el evento de actualizar, solo obtener todos los datos que se encuentra en la vista, así como los modificados, Y enviara todos los datos a un método de actualizar que se de tipo bool, para así luego verificar en el evento que todo halla salido correcto y redirigir a la pagina de todos los docentes con su respectivo mensaje.

En el método actualizar, va tener las mismas 4 condicionales que en el insert, en la primera solo actualizara sin tener en cuenta la imagen y el archivo en el segundo, actualizara teniendo encuentra todos los registros con la imagen, tiene una condición anidada donde verifica si el usuario tenia una imagen antes, si no solo la inserta llamando el método insertar imagen, si ya tenia una imagen antes, hará una consulta a ese docente, tomara el dato del nombre de la imagen, la buscara en el servidor y la eliminara, para luego insertar la nueva imagen, para la tercera condición hará lo mismo que la anterior pero teniendo en cuenta todos los datos y el archivo, y en la última condición tendrá en cuenta todos los registros incluyendo imagen y archivo cada una con sus misma validaciones verificando archivos anteriores para eliminarlos, en la plantilla de consultado se hace el mismo evento y se llama un objeto de la clase donde están todos los docentes, para invocar el método de actualizar.

Para los botones de eliminar imagen o archivo, se obtiene el código del docente y unos label para mostrar, y se llama a otro método para eliminar el archivo del registro, el método de tipo bool, para así después verificar si se redirige o no la página, en los métodos hará una consulta al docente verificando que si tenga el archivo registrado para eliminarlo de lo contrario avisará que no existe ningún archivo en el registro actual.

El mensaje se manejará con la misma variable de sesión y al cargar la pagina validara que valor tiene esta variable para así mostrarle al usuario.

### Evento Actualizar plantilla de todos los docentes:

```
//Actualizar
0 referencias
protected void LinkButton1_Click(object sender, EventArgs e)
{
    DropDownList usuCodigo = FormView1.FindControl("usuCodigoDropDownList1") as DropDownList;
    Label docCodigo = FormView1.FindControl("docCodigoLabel1") as Label;
    TextBox doTelefono = FormView1.FindControl("doTelefonoTextBox") as TextBox;
    TextBox docIngreso = FormView1.FindControl("docIngresoTextBox") as TextBox;
    TextBox docApellido1 = FormView1.FindControl("docApellido1TextBox") as TextBox;
    TextBox docApellido2 = FormView1.FindControl("docApellido2TextBox") as TextBox;
    TextBox docNombre1 = FormView1.FindControl("docNombre1TextBox") as TextBox;
    TextBox docNombre2 = FormView1.FindControl("docNombre2TextBox") as TextBox;
    TextBox docImagen = FormView1.FindControl("docImagenFileUpload1") as FileUpload;
    FileUpload docArchivo = FormView1.FindControl("docArchivoFileUpload1") as FileUpload;
    Label label9 = FormView1.FindControl("Label9") as Label;
    Label label10 = FormView1.FindControl("Label10") as Label;
    Label label11 = FormView1.FindControl("Label11") as Label;
    bool redirigir = false;
    redirigir = actualizar(usuCodigo, docCodigo, doTelefono, docIngreso, docApellido1, docApellido2, docNombre1, docNombre2, docImagen, docArchivo, label9, label10, label11);
    if (redirigir == true)
    {
        Response.Redirect("Adocentes.aspx");
    }
}
```

### Evento Actualizar plantilla de docente consultado:

```
0 referencias
protected void LinkButton1_Click(object sender, EventArgs e)
{
    DropDownList usuCodigo = FormView1.FindControl("usuCodigoDropDownList1") as DropDownList;
    Label docCodigo = FormView1.FindControl("docCodigoLabel1") as Label;
    TextBox doTelefono = FormView1.FindControl("doTelefonoTextBox") as TextBox;
    TextBox docIngreso = FormView1.FindControl("docIngresoTextBox") as TextBox;
    TextBox docApellido1 = FormView1.FindControl("docApellido1TextBox") as TextBox;
    TextBox docApellido2 = FormView1.FindControl("docApellido2TextBox") as TextBox;
    TextBox docNombre1 = FormView1.FindControl("docNombre1TextBox") as TextBox;
    TextBox docNombre2 = FormView1.FindControl("docNombre2TextBox") as TextBox;
    FileUpload docImagen = FormView1.FindControl("docImagenFileUpload1") as FileUpload;
    FileUpload docArchivo = FormView1.FindControl("docArchivoFileUpload1") as FileUpload;
    Label label9 = FormView1.FindControl("Label9") as Label;
    Label label10 = FormView1.FindControl("Label10") as Label;
    Label label11 = FormView1.FindControl("Label11") as Label;
    Adocentes actualiza = new Adocentes();
    bool redirigir = false;
    redirigir = actualizar(actualizar(usuCodigo, docCodigo, doTelefono, docIngreso, docApellido1, docApellido2, docNombre1, docNombre2, docImagen, docArchivo, label9, label10, label11));
    if (redirigir == true)
    {
        Response.Redirect("Adocentes.aspx");
    }
}
```

### Método Actualizar:

```
0 referencias
public bool actualizar(DropDownList usuCodigo, Label docCodigo, TextBox doTelefono, TextBox docIngreso, TextBox docApellido1, TextBox docApellido2, TextBox docNombre1, TextBox docNombre2, FileUpload docImagen, FileUpload docArchivo)
{
    bool retorno= false;
    String Imagen= "";
    String archivoVar= "";
    if (docImagen.FileName.Equals("") && docArchivo.FileName.Equals(""))
    {
        label9.Text = "";
        label10.Text = "";
        label11.Text = "";
        conexion.Open();
        string cadena = "UPDATE tbldocentes SET usuCodigo = "+usuCodigo.Text+", docApellido1 = "+docApellido1.Text+", docApellido2 = "+docApellido2.Text+", docNombre1 = "+docNombre1.Text+", docNombre2 = "+docNombre2.Text+",";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        comando.ExecuteReader();
        comando.Close();
        Session["actualizadoDoc"] = "actualizado";
        retorno = true;
    }
    else if (!docImagen.FileName.Equals("") && docArchivo.FileName.Equals(""))
    {
        label9.Text = "";
        label10.Text = "";
        label11.Text = "";
        imagenVar = docImagen.FileName;
        label9.Text = subirImagen(imagenVar, docImagen);
        if (label9.Text.Equals(nombreImagen))
        {
            string varImagen = "";
            conexion.Open();
            string cadena = "SelectImagen from tbldocentes WHERE docCodigo = "+docCodigo.Text+"";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            SqlDataReader registros = comando.ExecuteReader();
            while (registros.Read())
            {
                if ((registros["Imagen"].ToString()).Equals(""))
                {
                    varImagen = (registros["Imagen"].ToString());
                }
            }
            conexion.Close();
            if (varImagen.Equals(""))
            {
                conexion.Open();
                string cadena2 = "UPDATE tbldocentes SET usuCodigo = "+usuCodigo.Text+", docApellido1 = "+docApellido1.Text+", docApellido2 = "+docApellido2.Text+", docNombre1 = "+docNombre1.Text+", docNombre2 = "+docNombre2.Text+",";
                SqlCommand comando2 = new SqlCommand(cadena2, conexion);
                comando2.ExecuteReader();
                comando2.Close();
                Session["actualizadoDoc"] = "actualizado";
                retorno = true;
            }
        }
    }
}
```

```

        }
        else
        {
            //borrar imagen anterior del directorio
            //Dado el caso, verifico que exista el archivo..
            if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + varImagen))
            {
                System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + varImagen);
            }
            conexion.Open();
            string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
            SqlCommand comando2 = new SqlCommand(cadena2, conexion);
            comando2.ExecuteReader();
            conexion.Close();
            Session["guardadoDoc"] = "actualizado";
            retorno = true;
        }
    }
    else if (docImagen.FileName.Equals("") && !docArchivo.FileName.Equals(""))
    {
        label19.Text = "";
        label18.Text = "";
        label11.Text = "";
        archivoVar = docArchivo.FileName;
        label19.Text = subirArchivo(archivoVar, docArchivo);
        if (label9.Text.Equals(nombreArchivo))
        {
            string varArchivo = "";
            conexion.Open();
            string cadena = $"select archivo from tbldocumentes WHERE doccodigo = {docCodigo.Text}";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            SqlDataReader registros = comando.ExecuteReader();
            while (registros.Read())
            {
                if (!registros["archivo"].ToString().Equals(""))
                {
                    varArchivo = (registros["archivo"].ToString());
                }
            }
            conexion.Close();
            if (varArchivo.Equals(""))
            {
                conexion.Open();
                string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
                SqlCommand comando2 = new SqlCommand(cadena2, conexion);
                comando2.ExecuteReader();
                conexion.Close();
                Session["guardadoDoc"] = "actualizado";
                retorno = true;
            }
        }
        else
        {
            //borrar archivo anterior del directorio
            //Dado el caso, verifico que exista el archivo..
            if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + varArchivo))
            {
                System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + varArchivo);
            }
            conexion.Open();
            string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
            SqlCommand comando2 = new SqlCommand(cadena2, conexion);
            comando2.ExecuteReader();
            conexion.Close();
            Session["guardadoDoc"] = "actualizado";
            retorno = true;
        }
    }
    else if (docImagen.FileName.Equals("") && !docArchivo.FileName.Equals(""))
    {
        label19.Text = "";
        label18.Text = "";
        label11.Text = "";
        imagenesVar = docImagen.FileName;
        archivoVar = docArchivo.FileName;
        label19.Text = subirImagen(imagenesVar, docImagen);
        condicion2 = true;
        label10.Text = subirArchivo(archivoVar, docArchivo);
        if (label9.Text.Equals(nombreArchivo) && label10.Text.Equals(nombreArchivo))
        {
            string varArchivo = "";
            string varImagen = "";
            conexion.Open();
            string cadena = $"select imagen, archivo from tbldocumentes WHERE doccodigo = {docCodigo.Text}";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            SqlDataReader registros = comando.ExecuteReader();
            while (registros.Read())
            {
                if (!registros["archivo"].ToString().Equals(""))
                {
                    varArchivo = (registros["archivo"].ToString());
                }
                if (!registros["imagen"].ToString().Equals(""))
                {
                    varImagen = (registros["imagen"].ToString());
                }
            }
            conexion.Close();
            if (varImagen.Equals("") && varArchivo.Equals(""))
            {
                conexion.Open();
                string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
                SqlCommand comando2 = new SqlCommand(cadena2, conexion);
                comando2.ExecuteReader();
                conexion.Close();
                Session["guardadoDoc"] = "actualizado";
                retorno = true;
            }
            else if (varImagen.Equals("") && varArchivo.Equals(""))
            {
                if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + varImagen))
                {
                    System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + varImagen);
                }
                conexion.Open();
                string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
                SqlCommand comando2 = new SqlCommand(cadena2, conexion);
                comando2.ExecuteReader();
                conexion.Close();
                Session["guardadoDoc"] = "actualizado";
                retorno = true;
            }
            else if (varImagen.Equals("") && !varArchivo.Equals(""))
            {
                if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + varArchivo))
                {
                    System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + varArchivo);
                }
                conexion.Open();
                string cadena2 = $"UPDATE tbldocumentes SET usuCodigo = {usuCodigo.Text}, docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}'";
                SqlCommand comando2 = new SqlCommand(cadena2, conexion);
                comando2.ExecuteReader();
                conexion.Close();
                Session["guardadoDoc"] = "actualizado";
                retorno = true;
            }
            else if (!varImagen.Equals("") && !varArchivo.Equals(""))
            {
                if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + varImagen))
                {
                    System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + varImagen);
                }
                if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + varArchivo))
                {
                    System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + varArchivo);
                }
            }
        }
    }
}

```

```

        }
        conexion.Open();
        string cadena2 = $"UPDATE tbldocentes SET usuCodigo = '{usuCodigo.Text}', docApellido1 = '{docApellido1.Text}', docApellido2 = '{docApellido2.Text}', docNombre1 = '{docNombre1.Text}', docNombre2 = '{docNombre2.Text}' WHERE usuCodigo = '{usuCodigo2.Text}'";
        SqlCommand comando2 = new SqlCommand(cadena2, conexion);
        comando2.ExecuteNonQuery();
        conexion.Close();
        Session["guardadoDoc"] = "actualizado";
        retorno = true;
    }

    if (label9.Text.Equals(nombreImagen) && !label10.Text.Equals(nombreArchivo))
    {
        label9.Text = "";
        if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + nombreImagen))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + nombreImagen);
        }
    }
    if (label9.Text.Equals(nombreImagen) && label10.Text.Equals(nombreArchivo))
    {
        label10.Text = "";
        if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + nombreArchivo))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + nombreArchivo);
        }
    }
}
return retorno;
}

```

Evento eliminar foto del registro plantilla todos los docentes:

```

//Eliminar foto del registro
0 referencias
protected void LinkButton2_Click(object sender, EventArgs e)
{
    Label docCodigo = FormView1.FindControl("doccodigoLabel1") as Label;
    Label label9 = FormView1.FindControl("Label9") as Label;
    Label label10 = FormView1.FindControl("Label10") as Label;
    Label label11 = FormView1.FindControl("Label11") as Label;
    bool redirigir;
    redirigir = eliminarImgRegi(docCodigo, label9, label10, label11);
    if (redirigir == true)
    {
        Response.Redirect("Acdocentes.aspx");
    }
}

```

Evento eliminar foto del registro plantilla docente consultado:

```

0 referencias
protected void LinkButton2_Click(object sender, EventArgs e)
{
    Label docCodigo = FormView1.FindControl("doccodigoLabel1") as Label;
    Label label9 = FormView1.FindControl("Label9") as Label;
    Label label10 = FormView1.FindControl("Label10") as Label;
    Label label11 = FormView1.FindControl("Label11") as Label;
    Acdocentes eliminarImg = new Acdocentes();
    bool redirigir;
    redirigir = eliminarImg.eliminarImgRegi(docCodigo, label9, label10, label11);
    if (redirigir == true)
    {
        Response.Redirect("Acdocentes.aspx");
    }
}

```

## Método de eliminar foto:

```
2 referencias
public bool eliminarImgRegi(Label docCodigo, Label label9, Label label10, Label label11)
{
    bool retorno = false;
    String nombreImagen2 = "";
    label9.Text = "";
    label10.Text = "";
    label11.Text = "";
    conexion.Open();
    string cadena = $"select imagen, archivo from tbldocentes WHERE doccodigo = {docCodigo.Text}";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    SqlDataReader registros = comando.ExecuteReader();
    while (registros.Read())
    {
        if (!registros["imagen"].ToString().Equals(""))
        {
            nombreImagen2 = (registros["imagen"].ToString());
        }
    }
    conexion.Close();
    if (!nombreImagen2.Equals(""))
    {
        if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + nombreImagen2))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/imagenes/" + nombreImagen2);
        }
        conexion.Open();
        string cadena2 = $"UPDATE tbldocentes SET imagen = null WHERE doccodigo = {docCodigo.Text}";
        SqlCommand comando2 = new SqlCommand(cadena2, conexion);
        comando2.ExecuteReader();
        conexion.Close();
        Session["guardadoDoc"] = "actualizadoImgBorrado";
        retorno = true;
    }
    else
    {
        label9.Text = "No existe ninguna foto insertada en este registro";
    }
    return retorno;
}
```

## Evento eliminar archivo del registro:

```
//Eliminar archivo del registro
0 referencias
protected void LinkButton3_Click(object sender, EventArgs e)
{
    Label docCodigo = FormView1.FindControl("doccodigoLabel1") as Label;
    Label label9 = FormView1.FindControl("Label9") as Label;
    Label label10 = FormView1.FindControl("Label10") as Label;
    Label label11 = FormView1.FindControl("Label11") as Label;
    bool redirigir;
    redirigir = eliminarArcRegi(docCodigo, label9, label10, label11);
    if (redirigir == true)
    {
        Response.Redirect("Acdocentes.aspx");
    }
}
```

## Método eliminar archivo:

```
2 referencias
public bool eliminarArcRegi(Label docCodigo, Label label9, Label label10, Label label11)
{
    bool retorno = false;
    String nombreArchivo2 = "";
    label9.Text = "";
    label10.Text = "";
    label11.Text = "";
    conexion.Open();
    string cadena = $"select imagen, archivo from tbldocentes WHERE doccodigo = {docCodigo.Text}";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    SqlDataReader registros = comando.ExecuteReader();
    while (registros.Read())
    {
        if (!registros["archivo"].ToString().Equals(""))
        {
            nombreArchivo2 = (registros["archivo"].ToString());
        }
    }
    conexion.Close();
    if (!nombreArchivo2.Equals(""))
    {
        if (System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + nombreArchivo2))
        {
            System.IO.File.Delete(Server.MapPath(".") + "/archivos/" + nombreArchivo2);
        }
        conexion.Open();
        string cadena2 = $"UPDATE tbldocentes SET archivo = null WHERE doccodigo = {docCodigo.Text}";
        SqlCommand comando2 = new SqlCommand(cadena2, conexion);
        comando2.ExecuteReader();
        conexion.Close();
        Session["guardadoDoc"] = "actualizadoArcBorrado";
        retorno = true;
    }
    else
    {
        label9.Text = "No existe ningun archivo insertado en este registro";
    }
}
return retorno;
}
```

Al cargar la página:

```

else if (Session["guardadoDoc"].ToString().Equals("actualizado"))
{
    Label7.Text = "Se a actualizado con exito el registro";
    Label7.Visible = true;
    Button4.Visible = true;
    Session["guardadoDoc"] = "";
}
else if (Session["guardadoDoc"].ToString().Equals("actualizadoImgBorrado"))
{
    Label7.Text = "Se a eliminado la foto del registro con exito";
    Label7.Visible = true;
    Button4.Visible = true;
    Session["guardadoDoc"] = "";
}
else if (Session["guardadoDoc"].ToString().Equals("actualizadoArcBorrado"))
{
    Label7.Text = "Se a eliminado el archivo del registro con exito";
    Label7.Visible = true;
    Button4.Visible = true;
    Session["guardadoDoc"] = "";
}
}
```

Al seleccionar en el menú consultar galería de docentes mostrara una tabla de todos los docentes con las imágenes que tienen y el archivo con opción de ver los archivos:

The screenshot shows a web-based administration interface titled 'ADMINISTRADOR'. On the left, there's a sidebar with navigation links: 'Inicio', 'Administrador', 'Actualización de docentes', 'Cursos', 'Actualizar de estudiantes', and 'Imagenes y archivos'. The main content area displays a gridview with the following columns: Código Usuario, Código docente, Apellido 1 Docente, Apellido 2 Docente, Nombre 1 Docente, Nombre 2 Docente, Imagen, archivo, Foto, and PDF. Below the gridview, there's a copyright notice: 'Copyright © 2014-2020 System Software (SIRa). All rights reserved.' and a version number: 'Version 2.0'.

Código Usuario	Código docente	Apellido 1 Docente	Apellido 2 Docente	Nombre 1 Docente	Nombre 2 Docente	Imagen	archivo	Foto	PDF
3	49339877	Acer	Rivera	Carlos	Arturo	ejemplo.png	informe1.pdf		<a href="#">Abrir Archivo</a>
3	52322195	Acer	Rivera	Gladys	Elena	dos.png	informe2.pdf		<a href="#">Abrir Archivo</a>
3	55254515	Acosta	Alvarez	Edna	Fernanda	tres.png	informe3.pdf		<a href="#">Abrir Archivo</a>
3	58166834	Acosta	Ramon	Eduardo	Juan				<a href="#">Abrir Archivo</a>
3	61111113	Abaro	Abri	Claudia	Melina				<a href="#">Abrir Archivo</a>
3	64051472	Aleardo	Santamaría	Hanicy	Rocio				<a href="#">Abrir Archivo</a>
3	66963791	Avelia	Arroz	Dortz	Estela				<a href="#">Abrir Archivo</a>
3	69916110	Baron	Manrique	Luis	Carlos				<a href="#">Abrir Archivo</a>
3	72848429	Barrera	Cocunubo	Marsol					<a href="#">Abrir Archivo</a>
3	75780748	Barreto	Medina	Zulma	Rocio				<a href="#">Abrir Archivo</a>

### Explicación técnica para ver galería de docentes:

Se agrego un web form a la pagina maestra de administrador y aquí se insertó un gridview consultando la tabla de docentes, mostrando la imagen guardados en la carpeta imágenes tomando referencia el nombre de la imagen que esta insertada en la base de datos o así mismo para el archivo creando un hipervínculo para que se abra el archivo que tiene el docente.

### Agregar docentes a cursos:

En esta pestaña se puede asignar a los docentes las materias que van a dictar, consultar cuantas materias tiene asignado cada docente, eliminar si un docente ya no va a dictar una materia, y consultar cuantos registros hay en la pagina actual

Para asignar un docente a una materia solo consta de seleccionar el docente y el curso que se le quiere asignar y darle insertar y mostrara un mensaje de que todo salió correcto.

**Agregar docentes a cursos**

	Codigo del curso	Codigo del docente	Nombre del curso	Intensidad horaria del curso	Docente	Codigo de usuario
Seleccionar	1	49389877	Ingles.	5	Acero Rivera Carlos Arturo	3
Seleccionar	6	49389877	Español.	4	Acero Rivera Carlos Arturo	3
Seleccionar	6	52322195	Español.	4	Acero Rivera Gladys Elena	3
Seleccionar	5	52322195	Cobol	6	Acero Rivera Gladys Elena	3
Seleccionar	7	52322195	Finanzas.	7	Acero Rivera Gladys Elena	3
Seleccionar	1	52322195	Ingles.	5	Acero Rivera Gladys Elena	3
Seleccionar	2	52322195	Ingles 2	2	Acero Rivera Gladys Elena	3
Seleccionar	3	52322195	C++	3	Acero Rivera Gladys Elena	3
Seleccionar	2	58186634	Ingles 2	2	Acosta Rincon Eduardo Juan	3
Seleccionar	6	61119153	Español.	4	Alfaro Abril Claudia Yolima	3

12

Docente:

Curso:

Los datos se guardaron correctamente

Si un docente ya tiene asignado una materia y se intenta insertar a la misma mostrara (ERROR, Clave repetida)

**Agregar docentes a cursos**

	Codigo del curso	Codigo del docente	Nombre del curso	Intensidad horaria del curso	Docente	Codigo de usuario
Seleccionar	1	49389877	Ingles.	5	Acero Rivera Carlos Arturo	3
Seleccionar	6	49389877	Español.	4	Acero Rivera Carlos Arturo	3
Seleccionar	6	52322195	Español.	4	Acero Rivera Gladys Elena	3
Seleccionar	5	52322195	Cobol	6	Acero Rivera Gladys Elena	3
Seleccionar	7	52322195	Finanzas.	7	Acero Rivera Gladys Elena	3
Seleccionar	1	52322195	Ingles.	5	Acero Rivera Gladys Elena	3
Seleccionar	2	52322195	Ingles 2	2	Acero Rivera Gladys Elena	3
Seleccionar	3	52322195	C++	3	Acero Rivera Gladys Elena	3
Seleccionar	2	58186634	Ingles 2	2	Acosta Rincon Eduardo Juan	3
Seleccionar	6	61119153	Español.	4	Alfaro Abril Claudia Yolima	3

12

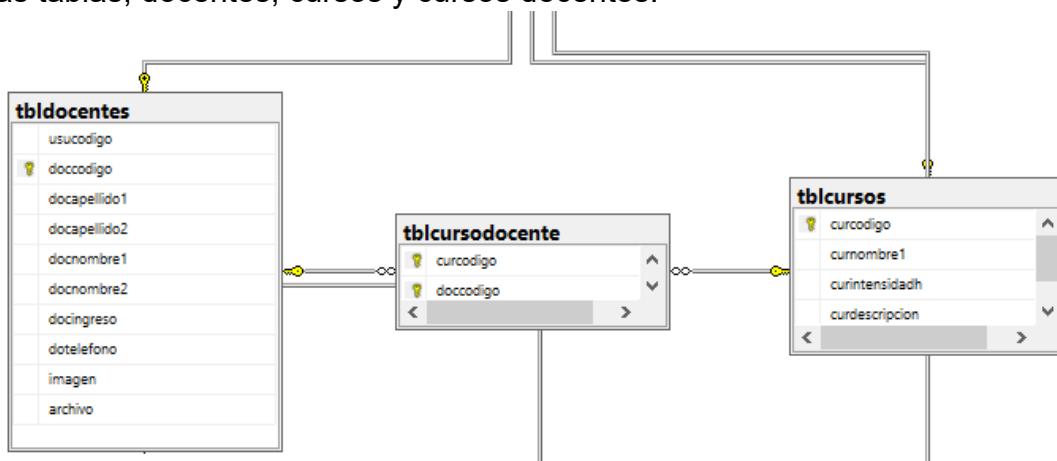
Docente:

Curso:

ERROR, clave duplicada

### Explicación técnica para asignar docentes a un curso:

Se crea un Gridview basado en una vista por la base de datos donde une las tablas, docentes, cursos y cursos docentes:



Para la asignación, se va insertar en la tabla tblcursodocente, para eso se generó un dropBox que traerá los datos de todos los docentes hechos por una vista en la base de datos para que muestre todo el nombre y el valor que obtendrá será el código del docente, para el curso también se generó un Dropbox donde su valor obtenido va a ser su código, de este modo para el evento Insertar se abre una conexión a la base de datos y se genera la sentencia SQL para inserta a la tabla tblcursosdocente, trayendo los códigos, y con el Try catch valida el error de llaves duplicadas:

```
0 referencias
protected void LinkButton1_Click(object sender, EventArgs e)
{
    conexion.Open();
    string docente = DropDownList1.SelectedValue;
    string curso = DropDownList2.SelectedValue;

    try
    {
        string cadena = $"INSERT INTO tblcursodocente (curcodigo,doccodigo) VALUES ({curso} , {docente});";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        comando.ExecuteNonQuery();
        Label5.Text = "Los datos se guardaron correctamente";
        GridView1.DataBind();
    }
    catch
    {
        Label5.Text = "ERROR, clave duplicada";
    }
    conexion.Close();
}
```

### Buscar cuantas materias tiene un docente:

Para buscar cuantas materias tiene asignado un docente solo consta de seleccionar el docente que se quiere consultar y mostrara el resultado después de oprimir el botón Buscar:

**Agregar docentes a cursos**

	Código del curso	Código del docente	Nombre del curso	Intensidad horaria del curso	Docente	Código de usuario
Seleccionar	1	64051472	Inglés.	5	Alvarado Santamaría Nancy Rocío	3
Seleccionar	1	66983791	Inglés.	5	Avella Arcoz Doriz Estela	3
Seleccionar	8	66983791	Liderazgo.	9	Avella Arcoz Doriz Estela	3
Seleccionar	5	69916110	Cobol	6	Baron Manrique Luis Carlos	3
Seleccionar	1	72848429	Inglés.	5	Barrera Cucunubá Marisol	3
Seleccionar	1	75780748	Inglés.	5	Barreto Medina Zulma Rocío	3
Seleccionar	2	78713067	Inglés 2	2	Becerra Avella Cyndy Jasbleidy	3
Seleccionar	3	81645386	C++	3	Becerra Becerra Llida Francisca	3
Seleccionar	5	84577705	Cobol	6	Bohórquez Rodríguez Jeffer	3

12

Docente:

Curso:

*El docente tiene asignado 1 materias*

**Agregar docentes a cursos**

	Código del curso	Código del docente	Nombre del curso	Intensidad horaria del curso	Docente	Código de usuario
Seleccionar	1	49389877	Inglés.	5	Acero Rivera Carlos Arturo	3
Seleccionar	6	49389877	Español.	4	Acero Rivera Carlos Arturo	3
Seleccionar	6	52322195	Español.	4	Acero Rivera Gladys Elena	3
Seleccionar	5	52322195	Cobol	6	Acero Rivera Gladys Elena	3
Seleccionar	7	52322195	Finanzas.	7	Acero Rivera Gladys Elena	3
Seleccionar	1	52322195	Inglés.	5	Acero Rivera Gladys Elena	3
Seleccionar	2	52322195	Inglés 2	2	Acero Rivera Gladys Elena	3
Seleccionar	3	52322195	C++	3	Acero Rivera Gladys Elena	3
Seleccionar	2	58186834	Inglés 2	2	Acosta Rincón Eduardo Juan	3
Seleccionar	6	61119153	Español.	4	Alfaró Abril Claudia Yolima	3

12

Docente:

Curso:

El docente no tiene materias asignadas

### Explicación técnica del evento buscar:

En este evento hace una búsqueda del docente con el código obtenido de el y con una variable contador y con un while se va incrementando mientras lee los registros, y con una condición valida si encontró materias asignadas a este docente o no

```
0 referencias
protected void LinkButton2_Click(object sender, EventArgs e)
{
    int cont = 0;
    conexion.Open();
    string cadena = $"SELECT curcodigo, doccodigo FROM tblcursodocente Where doccodigo = {this.DropDownList1.SelectedValue}";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    SqlDataReader registro = comando.ExecuteReader();
    while (registro.Read())
    {
        cont++;
    }
    if(cont == 0)
    {
        this.Label5.Text = "El docente no tiene materias asignadas";
    }
    if (cont > 0)
    {
        this.Label5.Text = $"El docente tiene asignado {cont} materias";
    }
    conexion.Close();
}
```

### Eliminar materias para un docente:

Para eliminar las materias asignadas a un docente seleccionar el docente y la materia que se le va a eliminar y darle al botón eliminar, y saldrá el aviso de que se borró exitosamente.

### Agregar docentes a cursos

	Codigo del curso	Codigo del docente	Nombre del curso	Intensidad horaria del curso	Docente	Codigo de usuario
Seleccionar	1	64051472	Ingles.	5	Alvarado Santamaria Nancy Rocio	3
Seleccionar	1	66983791	Ingles.	5	Avella Arcoz Doriz Estela	3
Seleccionar	8	66983791	Liderazgo.	9	Avella Arcoz Doriz Estela	3
Seleccionar	5	69916110	Cobol	6	Baron Manrique Luis Carlos	3
Seleccionar	1	72848429	Ingles.	5	Barrera Cocunubo Marisol	3
Seleccionar	1	75780748	Ingles.	5	Barreto Medina Zulma Rocio	3
Seleccionar	2	78713067	Ingles 2	2	Becerra Avella Cyndy Jasbleidy	3
Seleccionar	3	81645386	C++	3	Becerra Becerra LLda Francisca	3
Seleccionar	5	84577705	Cobol	6	Bohorquez Rodriguez Jeffer	3

12

Docente:

Curso:

### Agregar docentes a cursos

	Codigo del curso	Codigo del docente	Nombre del curso	Intensidad horaria del curso	Docente	Codigo de usuario
Seleccionar	1	64051472	Ingles.	5	Alvarado Santamaria Nancy Rocio	3
Seleccionar	1	66983791	Ingles.	5	Avella Arcoz Doriz Estela	3
Seleccionar	8	66983791	Liderazgo.	9	Avella Arcoz Doriz Estela	3
Seleccionar	1	72848429	Ingles.	5	Barrera Cocunubo Marisol	3
Seleccionar	1	75780748	Ingles.	5	Barreto Medina Zulma Rocio	3
Seleccionar	2	78713067	Ingles 2	2	Becerra Avella Cyndy Jasbleidy	3
Seleccionar	3	81645386	C++	3	Becerra Becerra LLda Francisca	3
Seleccionar	5	84577705	Cobol	6	Bohorquez Rodriguez Jeffer	3

12

Docente:

Curso:

Los datos se eliminaron con EXITO

### Explicación técnica para eliminar materia al docente:

En el evento se obtiene los códigos de docente y curso, y con la sentencia SQL utilizando el **where** con el operador **AND** se ejecuta la sentencia y con el try catch se controla la ejecución por si sale algún error:

```
0 referencias
protected void LinkButton3_Click(object sender, EventArgs e)
{
    conexion.Open();
    string docente = DropDownList1.SelectedValue;
    string curso = DropDownList2.SelectedValue;

    try
    {
        string cadena = $"DELETE FROM tblcursodocente WHERE curcodigo = {curso} AND doccodigo = {docente};";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        comando.ExecuteNonQuery();
        Label5.Text = "los datos se eliminaron con EXITO";
        GridView1.DataBind();
    }
    catch
    {
        Label5.Text = "ERROR, no se a podido eliminar";
    }
    conexion.Close();
}
```

## Contar materias de la página:

Solo consta de seleccionar la pagina y oprimir el botón contar pestañas, mostrara cuantas materias hay en página:

The screenshot shows a web page titled "Agregar docentes a cursos". Below the title is a table with columns: Código del curso, Código del docente, Nombre del curso, Intensidad horaria del curso, Docente, and Código de usuario. The table contains 8 rows of data. At the bottom of the table, there is a small number "12". Below the table, there is a form with fields for "Docente" (dropdown menu with value "Baron Manique Luis Carlos") and "Curso" (dropdown menu with value "Cobol"). There are two buttons: "Insertar" (green) and "Buscar" (green). To the right of the form, there is a yellow button labeled "Contar materias" with a blue oval around it. To the right of the "Contar materias" button, a message "Existen 8 materias en la pag" is displayed. Below the message is a red button labeled "Eliminar".

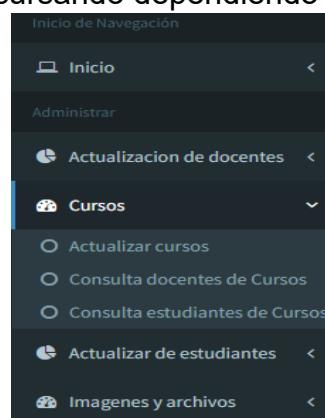
## Explicación técnica para contar registros de página:

En el evento atreves de un foreach y un contador se cuentas las filas del Gridview:

```
0 referencias
protected void Button2_Click(object sender, EventArgs e)
{
    int cont = 0;
    foreach (GridViewRow row in GridView1.Rows)
    {
        cont++;
    }
    Label5.Text = $"Existen {cont} materias en la pag";
}
```

## Cursos:

Para los cursos se tienen tres ítems, uno para crearlos o actualizarlos, otro para consultar los profesores que se tiene asignado y el otro para mirar los estudiantes que lo están cursando dependiendo del profesor:



En actualizar cursos se verá la tabla de los cursos que existen en el sistema, con las opciones de editar, seleccionar o eliminar, y en la parte inferior un botón por si se quiere crear un nuevo curso

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	Acciones
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>

[Agregar Curso](#)

### Insertar un curso nuevo:

Para insertar un curso nuevo, seleccionar agregar nuevo curso, llenar los campos y al guardar redirigirá la página donde están todos los cursos y mostrara un mensaje avisando que todo salió correcto:

### Insertar curso

**Nombre del curso:**

**Intensidad Horaria:**

**Descripción del curso:**

[Insertar](#) [Cancelar](#)

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
1009	Curso Nuevo	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>

[Agregar Curso](#)

Se a guardado con éxito el curso X

### Explicación técnica para insertar un nuevo curso:

Para la ventana de los cursos de agrego un GridView haciendo la consulta de la tabla de cursos, en un formulario se obtiene los valores ingresados por el usuario, y se genera la sentencia sql para insertar el nuevo registro, se asigna un valor a una nueva variable de sesión para mostrar el mensaje la cual será cargada por la clase donde muestra todos los cursos:

```
0 referencias
protected void LinkButton1_Click(object sender, EventArgs e)
{
    if (TextBox1.Text.Equals("") || TextBox2.Text.Equals("") || TextBox3.Text.Equals(""))
    {
        Label5.Text = "ERROR, debe insertar todos los campos";
    }
    else
    {
        conexion.Open();
        string cadena = $"INSERT INTO tblcursos (curnombre1, curintensidadh, curdescripcion) VALUES('{TextBox1.Text}', '{TextBox2.Text}', '{TextBox3.Text}')";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        comando.ExecuteReader();
        conexion.Close();
        Session["guardadoCur"] = "guardado";
        Response.Redirect("Accursos.aspx");
    }
}
```

```
0 referencias
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["guardadoCur"] != null)
    {
        if (Session["guardadoCur"].ToString().Equals("guardado"))
        {
            Label7.Text = "Se a guardado con éxito el curso";
            Label7.Visible = true;
            Button4.Visible = true;
            Session["guardadoCur"] = "";
        }
    }
}
```

### Editar un curso:

Para editar un curso seleccionar el botón editar y se habilitara los campos en la tabla para modificar los datos, darle actualizar y se evidenciara ahí mismo los cambios.

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
1009	Curso Nuevo Editado	4	Curso de formación corto 40 semanas	<a href="#">Actualizar</a> <a href="#">Cancelar</a>

[Agregar Curso](#)

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
1009	Curso Nuevo Editado	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>

[Agregar Curso](#)

**Eliminar un curso:**  
Oprimir el botón eliminar del curso que se quiere borrar, y se actualizara automáticamente la tabla:

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
1009	Curso Nuevo Editado	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>

[Agregar Curso](#)

Código del Curso	Nombre del curso	Intensidad horaria	Descripción	
1	Inglés.	5	Curso de formación corto 30 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
2	Inglés 2	2	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
3	C++	3	Curso de formación corto 60 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
4	Java	4	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
5	Cobol	6	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
6	Español.	4	Curso de formación corto 20 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
7	Finanzas.	7	Curso de formación corto 50 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>
8	Liderazgo.	9	Curso de formación corto 40 semanas	<a href="#">Editar</a> <a href="#">Seleccionar</a> <a href="#">Eliminar</a>

[Agregar Curso](#)

### Consultar docentes por cursos:

En esta pestaña se puede visualizar todos los docentes que tiene asignado un curso, para esto en la parte superior se tendrá un campo que mostrara los cursos que existen solo es seleccionarlo y la tabla se actualizara con los docentes que tiene el curso seleccionado:

The screenshot shows a dropdown menu titled "Listado de Cursos" containing the following options: C++, Inglés., Inglés 2, C++, Java, Cobol, Español., Finanzas., and Liderazgo. Below the dropdown is a table with two rows of data. The columns are labeled: Nombre del curso, Código del docente, Apellido 1, Apellido 2, Nombre 1, and Nombre 2. The first row has values: 3, C++, 52322195, Acero, Rivera, Gladys, and Elena. The second row has values: 3, C++, 81645386, Becerra, Becerra, Llida, and Francisa.

The screenshot shows a dropdown menu titled "Listado de Cursos" containing the option "Inglés.". Below the dropdown is a table with six rows of data. The columns are labeled: Código del curso:, Nombre del curso, Código del docente, Apellido 1, Apellido 2, Nombre 1, and Nombre 2. The rows contain the following data:

Seleccionar	1	Inglés.	49389877	Acero	Rivera	Carlos	Arturo
Seleccionar	1	Inglés.	52322195	Acero	Rivera	Gladys	Elena
Seleccionar	1	Inglés.	64051472	Alvarado	Santamaría	Nancy	Rocio
Seleccionar	1	Inglés.	66983791	Avella	Arcoz	Doriz	Estela
Seleccionar	1	Inglés.	72848429	Barrera	Cocunubo	Marisol	
Seleccionar	1	Inglés.	75780748	Barreto	Medina	Zulma	Rocio

### Explicación técnica para consultar los docentes de un curso:

Se crea un GridView con los datos de la tabla cursos docente en una vista, trayendo los datos del curso y el docente haciendo la busque a la tabla tblcursodocentes, con la sentencia sql where, obteniendo el valor de un dropBox que consulta los cursos que existen en el sistema y un método autoPostBack, que refresca la pagina con el valor seleccionado de esta forma permite la búsqueda dinámica.

### Consultar estudiantes de cursos:

En esta ventana se podrá consultar los estudiantes que están asignados a un curso y con su docente, solo hay que seleccionar el curso que se quiere consultar, en el campo de docentes mostrara los docentes que dictan ese curso, se selecciona y traerá los datos correspondientes:

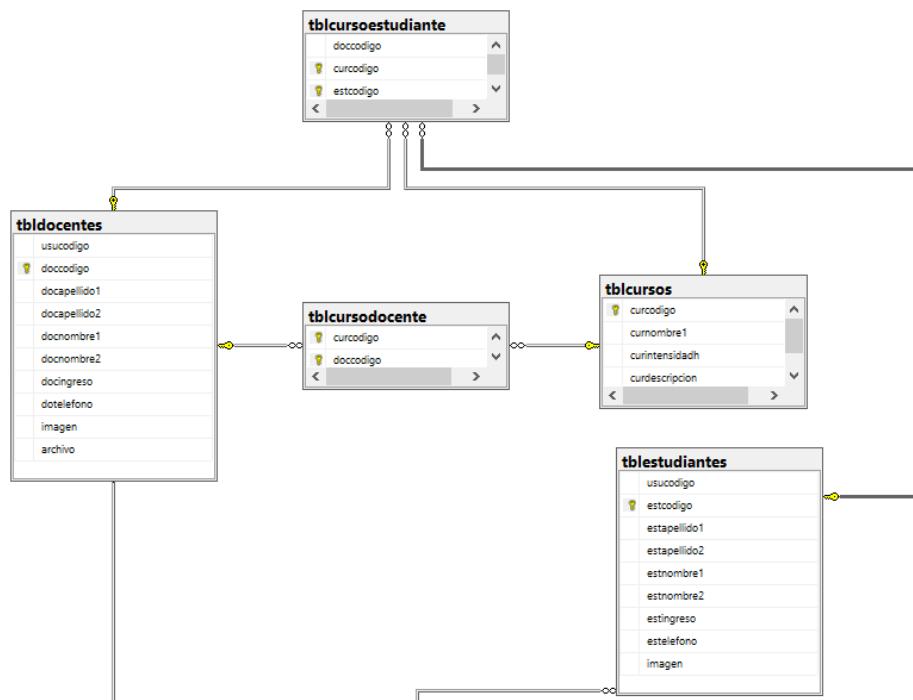
The screenshot shows a web application interface titled 'ADMINISTRADOR'. On the left, there's a sidebar with navigation links: 'Inicio', 'Administrador', 'Actualización de docentes', 'Cursos', 'Actualizar de estudiantes', and 'Imagenes y archivos'. The main content area has two dropdown menus: 'Listado de Cursos:' containing 'Inglés.', 'Inglés.', 'Inglés 2', 'C++', 'Java', 'Cobol', 'Español', 'Finanzas', and 'Liderazgo', with 'Español' selected; and 'Listado de docentes por curso:' containing 'Acero Rivera Carlos Arturo', with his name highlighted. Below these is a table with columns: 'Codigo del curso', 'Nombre del curso', 'Codigo del estudiante', and 'Estudiante'. It shows two rows: one for 'Inglés.' with student 'Rodrigo Gonzalez' and another for 'Inglés.' with student 'Julian Barreto'.

This screenshot shows the same application interface after selecting 'Español' from the course dropdown. The 'Listado de docentes por curso:' dropdown now displays a list of teachers: 'Acero Rivera Gladys Elena', 'Acero Rivera Carlos Arturo', 'Acero Rivera Gladys Elena', and 'Alfarro Abril Claudia Yolima', with 'Acero Rivera Gladys Elena' highlighted. The main content area is currently empty, indicating that the results table is still being populated or is empty.

Código del curso	Nombre del curso	Código del estudiante	Estudiante
6	Español.	14868215	juan alfonzo
6	Español.	15427855	santiago garcia
6	Español.	15483525	julian jaramillo
6	Español.	34927681	linda castillo
6	Español.	35489628	david avellaneda

### Explicación técnica consultar estudiantes de cursos:

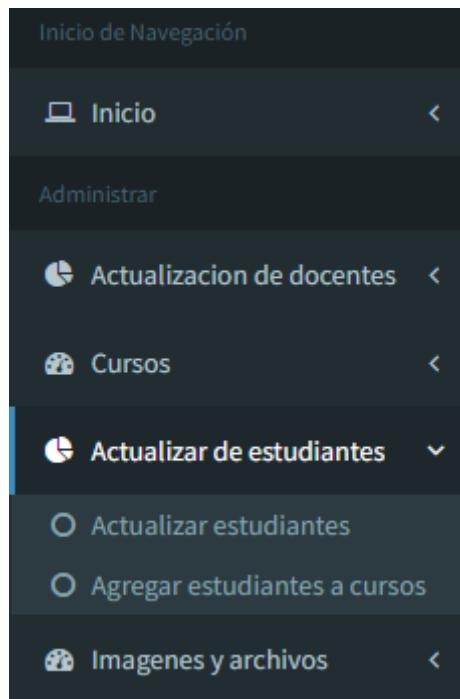
Para consultar esta tabla se creo una vista trayendo los datos de tblcursoestudiantes, los cursos, estudiantes y docentes.



Se agrega un dropBox que consulte los cursos que existe en el sistema, en otro dropBox consultara los docentes que están en el curso seleccionado, con sentencias SQL Where y el GridView la consulta será con la vista y la condición where utilizando el operador and con los valores código del curso y código del docente para que traiga los estudiantes correspondientes.

## Estudiantes:

Para los estudiantes se tiene los ítems de actualización y agregar estudiantes a cursos:



Para actualizar estudiantes es el mismo procedimiento que para los docentes:

A screenshot of the 'Administrador' application interface. The left sidebar shows the navigation menu with 'Actualizar de estudiantes' selected. The main content area is titled 'Actualización de estudiantes' and displays student details: Código de usuario: 4, Código de estudiante: 14868215, Ingreso: 10229360, Teléfono: 8; Apellido 1: alfonzo,Apellido2: castillo, Nombre 1: juan, Nombre 2: esteban. Below the form are three buttons: Editar, Eliminar, and Nuevo. A small watermark '12345678910...' is visible at the bottom of the form area.

Buscar Estudiante por Código:

### Crear estudiante

Código de usuario:	<input type="text" value="Estudiante"/>	Apellido 1:	<input type="text" value="Zabala"/>	Imagen:	<input type="button" value="Seleccionar archivo"/> Sin archivos seleccionados
Código de estudiante:	<input type="text" value="114"/>	Apellido 2:	<input type="text" value="Baez"/>		
Ingreso:	<input type="text" value="1516"/>	Nombre 1:	<input type="text" value="Vanyhony"/>		
Teléfono:	<input type="text" value="3203590398"/>	Nombre 2:	<input type="text" value="Esneyder"/>		

Buscar Estudiante por Código:

### Actualización de estudiantes

Código de usuario:	4	Apellido 1:	Zabala	Imagen:
Código de estudiante:	114	Apellido 2:	Baez	
Ingreso:	1516	Nombre 1:	Vanyhony	
Teléfono:	3203590398	Nombre 2:	Esneyder	

12345678910... x

Se ha guardado con éxito el registro

Buscar Estudiante por Código:

### Actualización de estudiantes

Código de usuario:	4	Apellido 1:	Zabala	Imagen:	ejemplo-1.png
Código de estudiante:	114	Apellido 2:	Baez		
Ingreso:	1516	Nombre 1:	Vanyhony Editado		
Teléfono:	3203590398	Nombre 2:	Esneyder		

12345678910... x

Se ha actualizado con éxito el registro

Buscar Estudiante por Código:

### Actualizar estudiante

Código de usuario:	<input type="text" value="Estudiante"/>	Apellido 1:	<input type="text" value="Zabala"/>	Imagen:	<input type="button" value="Seleccionar archivo"/> Sin arch...clonados
Código de estudiante:	<input type="text" value="114"/>	Apellido 2:	<input type="text" value="Baez"/>		
Ingreso:	<input type="text" value="1516"/>	Nombre 1:	<input type="text" value="Vanyhony Editado"/>		
Teléfono:	<input type="text" value="3203590398"/>	Nombre 2:	<input type="text" value="Esneyder"/>		

12345678910...

Buscar Estudiante por Código:

**Actualización de estudiantes**

Código de usuario:	4	Apellido 1:	Zabala	Imagen:
Código de estudiante:	114	Apellido 2:	Baez	
Ingreso:	1516	Nombre 1:	Vanyhony Editado	
Teléfono:	3203590398	Nombre 2:	Esneyder	

[Editar](#) [Eliminar](#) [Nuevo](#)

12345678910... X

Se a eliminado la foto del registro con éxito

Buscar Estudiante por Código:

**Actualización de estudiantes**

Código de usuario:	4	Apellido 1:	Zabala	Imagen:
Código de estudiante:	114	Apellido 2:	Baez	
Ingreso:	1516	Nombre 1:	Vanyhony Editado	
Teléfono:	3203590398	Nombre 2:	Esneyder	

[Editar](#) [Eliminar](#)

Buscar Estudiante por Código:

**Actualización de estudiantes**

Código de usuario:	4	Apellido 1:	alfonzo	Imagen:
Código de estudiante:	14868215	Apellido 2:	castillo	
Ingreso:	10229360	Nombre 1:	juan	
Teléfono:	8	Nombre 2:	esteban	

[Editar](#) [Eliminar](#) [Nuevo](#)

12345678910... X

Se a eliminado con éxito el registro

## Agregar estudiantes a cursos:

Para este proceso, es igual que asignar materias a docentes, la única diferencia es que aquí se tendrá en cuenta los docentes que están dictando el curso.

Elegir el curso que se va a inscribir el estudiante, luego los docentes disponibles para este curso y en estudiantes elegir el estudiante que se va a inscribir, luego seleccionar insertar, hará las respectivas validaciones el sistema, es decir que el estudiante no esté inscrito ya en este curso así sea con otro docente, y si todo es correcto se ingresa:

**Agregar estudiantes a cursos**

	Código del curso	Nombre del curso	Código del docente	Docente	Código del estudiante	Estudiante
Seleccionar	6	Español.	61119153	Claudia Alfaro	98364581	Cyndy garcia
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	2	Inglés 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Inglés.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Inglés.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Inglés.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Inglés.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	34927681	linda castillo

12

Curso:  Estudiante:  **ERROR, clave duplicada**

Docente:  **Insertar Buscar Eliminar Contar materias**

**Agregar estudiantes a cursos**

	Código del curso	Nombre del curso	Código del docente	Docente	Código del estudiante	Estudiante
Seleccionar	1	Inglés.	49389877	Carlos Acero	98364581	Cyndy garcia
Seleccionar	6	Español.	61119153	Claudia Alfaro	98364581	Cyndy garcia
Seleccionar	2	Inglés 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Inglés.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Inglés.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Inglés.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Inglés.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo

12

Curso:  Estudiante:  **Los datos se guardaron correctamente**

Docente:  **Insertar Buscar Eliminar Contar materias**

Si no hay ningún docente asignado a un curso aun y se intenta insertar el estudiante a este curso no lo dejará y dará error

**Agregar estudiantes a cursos**

Seleccionar	Codigo del curso	Nombre del curso	Codigo del docente	Docente	Codigo del estudiante	Estudiante
Seleccionar	<b>1</b>	Ingles.	<b>49389877</b>	Carlos Acero	<b>98364581</b>	Cyndy garcia
Seleccionar	6	Español.	61119153	Claudia Alfaro	98364581	Cyndy garcia
Seleccionar	2	Ingles 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Ingles.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Ingles.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Ingles.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Ingles.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo

12

Curso:  Estudiante:  **ERROR! Este curso no tiene ningun docente asignado aun**  
 Docente:

### Explicación técnica para insertar estudiantes a curso:

Se inserto un GridView consultado por una vista que muestra los docentes, cursos y estudiantes, con la tabla tblCursosEstudiantes.

Para agregar estudiantes se agrega un DropBox que traerá los datos de todos los cursos del sistema, en otro Dropbox traerá la consulta de docentes que hay por el curso seleccionado y en otro DropBox traerá a todos los estudiantes del sistema.

En el evento insertar primero se validará si existe un docente seleccionado si no dará error, si lo hay, obtendrá los valores de docente, curso, y estudiante, se hará la sentencia SQL para insertar los datos a la tabla tblCursoEstudiantes. Y si el código del curso y el estudiante se repiten se valida con un try catch

```
0 referencias
protected void LinkButton1_Click(object sender, EventArgs e)
{
    string curso = DropDownList1.SelectedValue;
    string docente = DropDownList2.SelectedValue;
    string estudiante = DropDownList3.SelectedValue;
    if (docente.Equals("")){
        Label5.Text = "ERROR! Este curso no tiene ningun docente asignado aun";
    }
    else
    {
        try
        {
            conexion.Open();
            string cadena = $"INSERT INTO tbcursoestudiante (doccodigo,curcodigo, estcodigo) VALUES ({docente}, {curso} , {estudiante})";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            comando.ExecuteNonQuery();
            Label5.Text = "Los datos se guardaron correctamente";
            GridView1.DataBind();
        }
        catch
        {
            Label5.Text = "ERROR, clave duplicada";
        }
        conexion.Close();
    }
}
```

## Buscar:

Lo que hace el botón buscar es contar cuantas materias tiene inscrito un estudiante, solo hay que seleccionar el estudiante y oprimir el botón buscar y el sistema dará el resultado:

	Código del curso	Nombre del curso	Código del docente	Docente	Código del estudiante	Estudiante
Seleccionar	1	Inglés.	49389877	Carlos Acero	98364581	Cyndy garcia
Seleccionar	6	Español.	61119153	Claudia Alfaro	98364581	Cyndy garcia
Seleccionar	2	Inglés 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Inglés.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Inglés.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Inglés.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Inglés.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo

12

Curso:	<input style="width: 150px;" type="text" value="Inglés."/>	Estudiante:	<input style="width: 150px;" type="text" value="alfonso castillo juan esteban"/>	El estudiante tiene inscrito 2 materias		
Docente:	<input style="width: 150px;" type="text" value="Acero Rivera Carlos Arturo"/>	<input style="width: 50px; background-color: #0070C0; color: white;" type="button" value="Insertar"/>	<input style="width: 50px; background-color: #00AEEF; color: white;" type="button" value="Buscar"/>	<input style="width: 50px; background-color: #FF0000; color: white;" type="button" value="Eliminar"/>	<input style="width: 100px; background-color: #FFFF00; color: black;" type="button" value="Contar materias"/>	

	Código del curso	Nombre del curso	Código del docente	Docente	Código del estudiante	Estudiante
Seleccionar	1	Inglés.	49389877	Carlos Acero	98364581	Cyndy garcia
Seleccionar	6	Español.	61119153	Claudia Alfaro	98364581	Cyndy garcia
Seleccionar	2	Inglés 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Inglés.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Inglés.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Inglés.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Inglés.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo

12

Curso:	<input style="width: 150px;" type="text" value="Inglés."/>	Estudiante:	<input style="width: 150px;" type="text" value="saenz paez claudia Rocio"/>	El estudiante no esta inscrito a una materia		
Docente:	<input style="width: 150px;" type="text" value="Acero Rivera Carlos Arturo"/>	<input style="width: 50px; background-color: #0070C0; color: white;" type="button" value="Insertar"/>	<input style="width: 50px; background-color: #00AEEF; color: white;" type="button" value="Buscar"/>	<input style="width: 50px; background-color: #FF0000; color: white;" type="button" value="Eliminar"/>	<input style="width: 100px; background-color: #FFFF00; color: black;" type="button" value="Contar materias"/>	

## Explicación técnica para buscar:

En el evento obtiene el código del estudiante y hace una búsqueda en la tabla tblcursoestudiante y con una variable de conteo y con un ciclo while da el resultado de cuantas veces se repite el estudiante en esta tabla:

```

0 referencias
protected void LinkButton2_Click(object sender, EventArgs e)
{
    int cont = 0;
    conexion.Open();
    string cadena = $"SELECT curcodigo, estcodigo FROM tblcursoestudiante Where estcodigo = {this.DropDownList3.SelectedValue};";
    SqlCommand comando = new SqlCommand(cadena, conexion);
    SqlDataReader registro = comando.ExecuteReader();
    while (registro.Read())
    {
        cont++;
    }
    if (cont == 0)
    {
        this.Label5.Text = "El estudiante no esta inscrito a una materia";
    }
    if (cont > 0)
    {
        this.Label5.Text = $"El estudiante tiene inscrito {cont} materias";
    }
    conexion.Close();
}

```

## Eliminar materia de estudiante:

Para eliminar una metería de un estudiante, es seleccionar la materia y el estudiante y darle eliminar, y dará como resultado un mensaje que salió todo correcto:

	Codigo del curso	Nombre del curso	Codigo del docente	Docente	Codigo del estudiante	Estudiante
Seleccionar	1	Ingles.	49389877	Carlos Acero	98364581	Cyndy garcia
Seleccionar	<b>6</b>	Español.	<b>61119153</b>	<b>Claudia Alfaro</b>	<b>98364581</b>	<b>Cyndy garcia</b>
Seleccionar	2	Ingles 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Ingles.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Ingles.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Ingles.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Ingles.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo

12

Curso: <input style="width: 150px;" type="text" value="Español."/>	Estudiante: <input style="width: 150px;" type="text" value="garcia Avella Cyndy Jasbleidy"/>	<b>El estudiante tiene inscrito 2 materias</b>
Docente: <input style="width: 150px;" type="text" value="Acero Rivera Carlos Arturo"/>	<b>Insertar</b> <b>Buscar</b> <b>Eliminar</b>	<b>Contar materias</b>

	Codigo del curso	Nombre del curso	Codigo del docente	Docente	Codigo del estudiante	Estudiante
Seleccionar	<b>1</b>	Ingles.	<b>49389877</b>	<b>Carlos Acero</b>	<b>98364581</b>	<b>Cyndy garcia</b>
Seleccionar	2	Ingles 2	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	6	Español.	49389877	Carlos Acero	35489628	david avellaneda
Seleccionar	1	Ingles.	75780748	Zulma Barreto	75396325	dayana torres
Seleccionar	1	Ingles.	66983791	Doriz Avella	14868215	juan alfonzo
Seleccionar	6	Español.	49389877	Carlos Acero	14868215	juan alfonzo
Seleccionar	1	Ingles.	49389877	Carlos Acero	75986335	julian Barreto
Seleccionar	1	Ingles.	72848429	Marisol Barrera	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	15483525	julian jaramillo
Seleccionar	6	Español.	49389877	Carlos Acero	34927681	linda castillo

12

Curso: <input style="width: 150px;" type="text" value="Español."/>	Estudiante: <input style="width: 150px;" type="text" value="garcia Avella Cyndy Jasbleidy"/>	<b>los datos se eliminaron con EXITO</b>
Docente: <input style="width: 150px;" type="text" value="Acero Rivera Carlos Arturo"/>	<b>Insertar</b> <b>Buscar</b> <b>Eliminar</b>	<b>Contar materias</b>

## Explicación técnica para eliminar materias de un estudiante:

El evento ejecuta una sentencia SQL delete, donde esté presente el código del curso y del estudiante, el evento esta controlado por un Try catch por si llega haber algún error:

```

protected void LinkButton3_Click(object sender, EventArgs e)
{
    conexion.Open();
    string estudiante = DropDownList3.SelectedValue;
    string curso = DropDownList1.SelectedValue;

    try
    {
        string cadena = $"DELETE FROM tblcursoestudiante WHERE curcodigo = {curso} AND estcodigo = {estudiante};";
        SqlCommand comando = new SqlCommand(cadena, conexion);
        comando.ExecuteNonQuery();
        Label5.Text = "los datos se eliminaron con EXITO";
        GridView1.DataBind();
    }
    catch
    {
        Label5.Text = "ERROR, no se a podido eliminar";
    }
    conexion.Close();
}

```

### Contar materias:

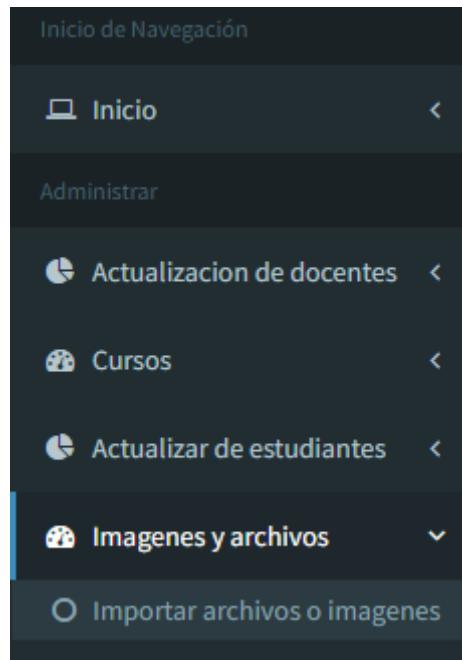
Hará el mismo procedimiento que en la tabla de asignación de cursos para docentes, oprimir el botón contar materias y dará resultado las materias que están en la página:

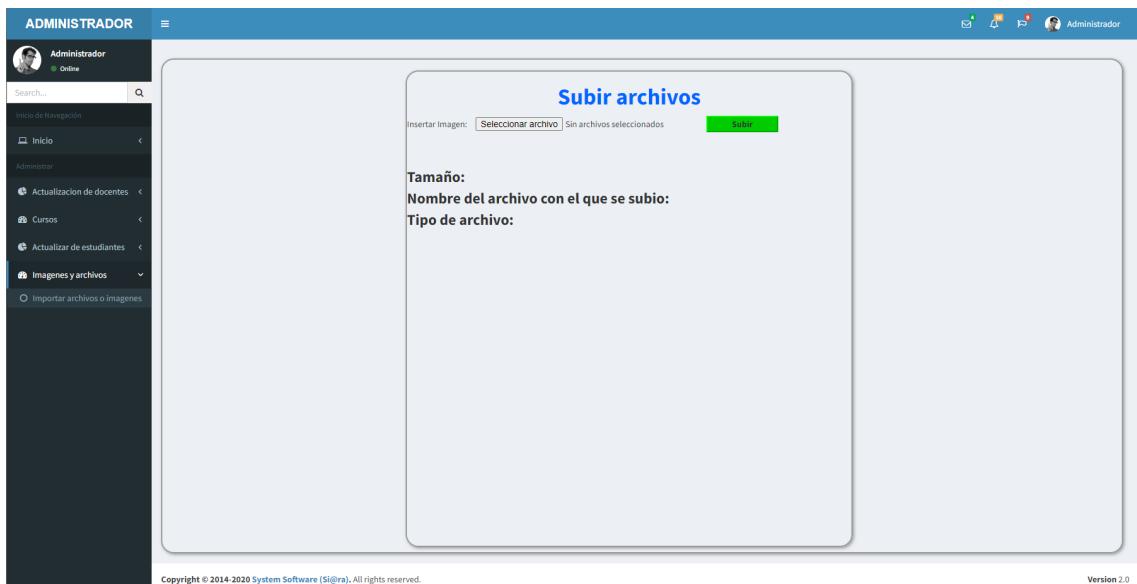
The screenshot shows a table titled "Agregar estudiantes a cursos" with columns: Código del curso, Nombre del curso, Código del docente, Docente, Código del estudiante, and Estudiante. The table lists various student records. At the bottom of the table, there is a footer section with dropdown menus for Curso (Inglés), Estudiante (Alfonzo Castillo Juan Esteban), Docente (Acero Rivera Carlos Arturo), and buttons for Insertar, Buscar, Eliminar, and Contar materias. The "Contar materias" button is circled in red.

	Código del curso	Nombre del curso	Código del docente	Docente	Código del estudiante	Estudiante
Seleccionar	5	Cobol	84577705	Jeffer Bohorquez	34927681	linda castillo
Seleccionar	7	Finanzas.	49389877	Carlos Acero	79286145	Maicol Barrera
Seleccionar	8	Liderazgo.	66983791	Doriz Avella	69916110	maria gutierrez
Seleccionar	1	Inglés.	49389877	Carlos Acero	45123548	Rodrigo Gonzalez
Seleccionar	1	Inglés.	64051472	Nancy Alvarado	15427855	santiago garcia
Seleccionar	2	Inglés 2	58186834	Eduardo Acosta	15427855	santiago garcia
Seleccionar	6	Español.	49389877	Carlos Acero	15427855	santiago garcia
Seleccionar	2	Inglés 2	78713067	Cyndy Becerra	96522746	yeferzon Bohorquez

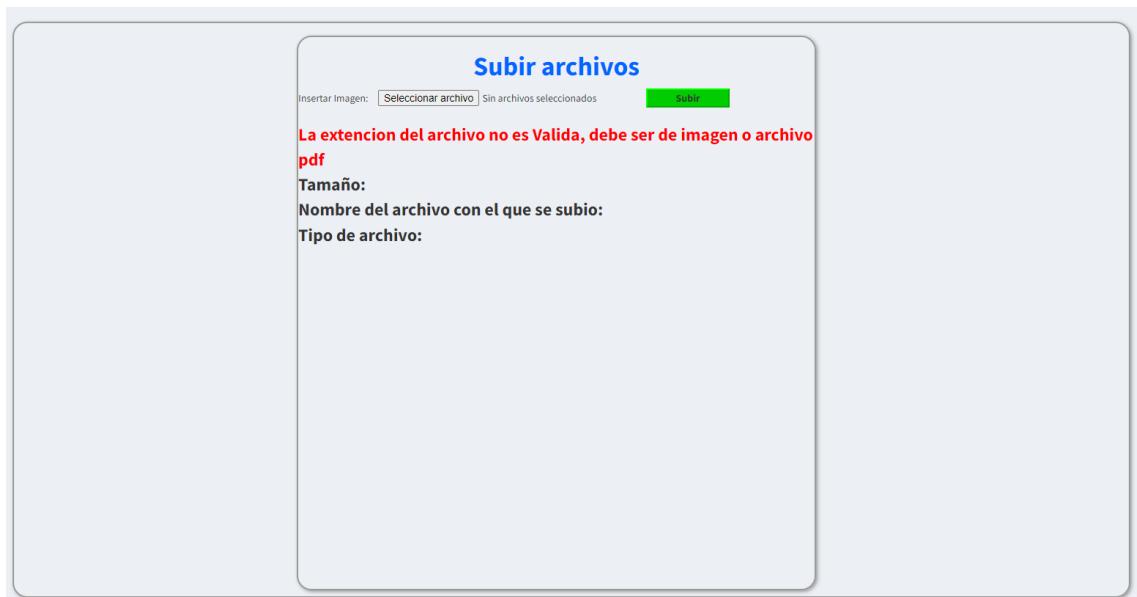
### Subir imágenes y archivos

En el menú estará la opción imágenes y archivos al desplegar dará un ítem que dice importar archivos o imágenes, aquí se podrá subir archivos o imágenes al servidor:





Para subir el archivo tener en cuenta que el peso máximo es de 200 kb, y debe ser extensión de imágenes o pdf, si estas condiciones no se cumplen dará un error.



Al subir una imagen, si todo sale correcto mostrara la información de la imagen y la imagen subida.

**Subir archivos**

Insertar Imagen:  Sin archivos seleccionados

**La imagen se a subido correctamente**

Tamaño: 0,006621 MB

Nombre del archivo con el que se subio: ejemplo.png

Tipo de archivo: .png

**Imagen subida:**



En caso de que ya existe , la imagen se guardara con el nombre que se subió agregando un – y un numero consecutivo.

**Subir archivos**

Insertar Imagen:  Sin archivos seleccionados

**La imagen se a subido correctamente**

Tamaño: 0,006621 MB

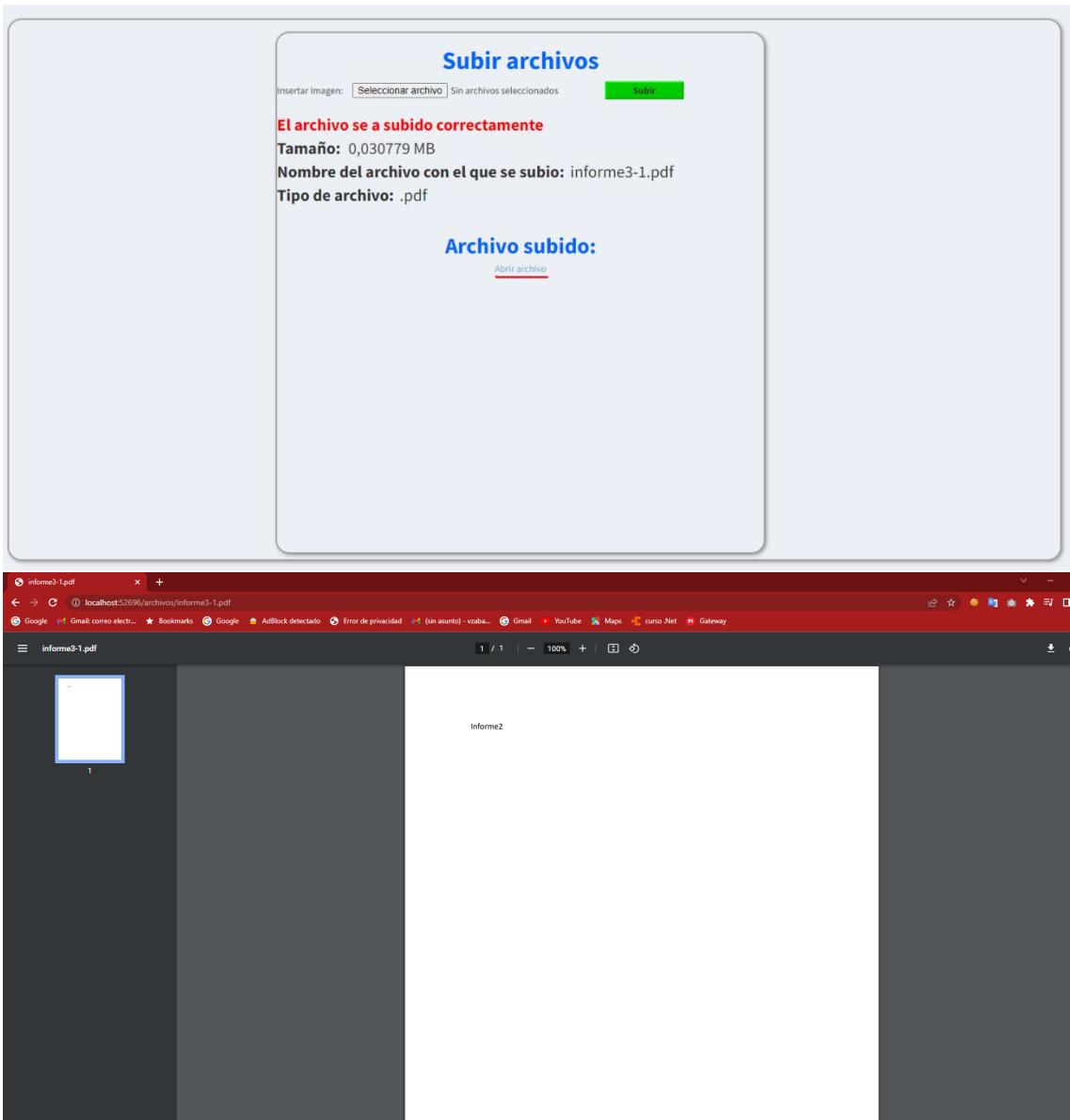
Nombre del archivo con el que se subio: ejemplo-1.png

Tipo de archivo: .png

**Imagen subida:**



Para los archivos, será lo mismo, obtendrá la información del documento y un enlace para abrirlo



### Explicación técnica para la subida de archivos:

En el evento hace las verificaciones con condicionales de la extensión del archivo, luego verifica el peso en bytes, convertido a kb, luego verifica si existe el archivo ya, y si es así, le agrega con un contador un numero consecutivo y lo guarda en el servidor y se asigna todos los valores necesarios a los label.

```

0 referencias
protected void Button2_Click(object sender, EventArgs e)
{
    ext = System.IO.Path.GetExtension(FileUpload1.FileName);
    String nombreArchivo = FileUpload1.FileName;
    int contador = 0;
    bool condicion = true;
    if(ext.Equals(".bmp") || ext.Equals(".gif") || ext.Equals(".tif") || ext.Equals(".png") || ext.Equals(".pdf"))
    {
        while (condicion == true)
        {
            if (System.IO.File.Exists(Server.MapPath(".") + "/imagenes/" + nombreArchivo) || System.IO.File.Exists(Server.MapPath(".") + "/archivos/" + nombreArchivo))
            {
                string variableNombreSinExt = System.IO.Path.GetFileNameWithoutExtension(FileUpload1.FileName);
                contador++;
                nombreArchivo = $"{variableNombreSinExt}-{contador}{ext}";
            }
            else
            {
                condicion = false;
            }
        }
        if (FileUpload1.PostedFile.ContentLength > 200000)
        {
            tamanoMb = ((double)FileUpload1.PostedFile.ContentLength / (double)1000) / 1000;
            Label7.Text = $"{tamanoMb} MB";
            Label5.Text = "Se supero el peso de subida para la imagen, el peso debe ser menor de 0.2mb (200kb)";
            Label9.Text = "";
            Label11.Text = "";
            Label12.Text = "";
            Image1.ImageUrl = "";
            HyperLink1.Text = "";
            HyperLink1.NavigateUrl = "";
        }
        else if (ext.Equals(".bmp") || ext.Equals(".gif") || ext.Equals(".tif") || ext.Equals(".png"))
        {
            tamanoMb = ((double)FileUpload1.PostedFile.ContentLength / (double)1000) / 1000;
            FileUpload1.SaveAs(Server.MapPath(".") + "/imagenes/" + nombreArchivo); //guarda el archivo contenido en FileUpload1
            Label7.Text = $"{tamanoMb} MB";
            Label9.Text = nombreArchivo;
            Label5.Text = "La imagen se a subido correctamente";
            Label11.Text = ext;
            Label12.Text = "Imagen subida:";
            Image1.ImageUrl = $"~/imagenes/{nombreArchivo}";
            HyperLink1.Text = "";
            HyperLink1.NavigateUrl = "";
        }
        else if (ext.Equals(".pdf"))
        {
            tamanoMb = ((double)FileUpload1.PostedFile.ContentLength / (double)1000) / 1000;
            FileUpload1.SaveAs(Server.MapPath(".") + "/archivos/" + nombreArchivo); //guarda el archivo contenido en FileUpload1
            Label7.Text = $"{tamanoMb} MB";
            Label9.Text = nombreArchivo;
            Label5.Text = "El archivo se a subido correctamente";
            Label11.Text = ext;
            Label12.Text = "Archivo subido:";
            HyperLink1.Text = "Abrir archivo";
            HyperLink1.NavigateUrl = $"~/archivos/{nombreArchivo}";
            Image1.ImageUrl = "";
        }
    }
    else
    {
        Label11.Text = ext;
        Label5.Text = "La extencion del archivo no es Valida, debe ser de imagen o archivo pdf";
        Label7.Text = "";
        Label9.Text = "";
        Label11.Text = "";
        Label12.Text = "";
        Image1.ImageUrl = "";
        HyperLink1.Text = "";
        HyperLink1.NavigateUrl = "";
    }
}
}

```

Al cerrar sesión se enviará a la página de logueo.

**ADMINISTRADOR**

Página de inicio Administrador

Cerrar Sesión

**Proyecto**

Ingresa a tu cuenta

Seleciona tu usuario & contraseña para ingresar

Usuario: Administrador

Contraseña:

Remember me

Login

Por vanyhony Esneyder Zabala Baez

### Logueo para docente:

Seleccionar el usuario docente y ingresar la contraseña que es el campo de ingreso con el que se registró, si no es correcto se mostrara el aviso y volverá a la pagina principal

**Proyecto ASP**

## Ingrésa a tu cuenta

Selecciona tu usuario & contraseña para ingresar

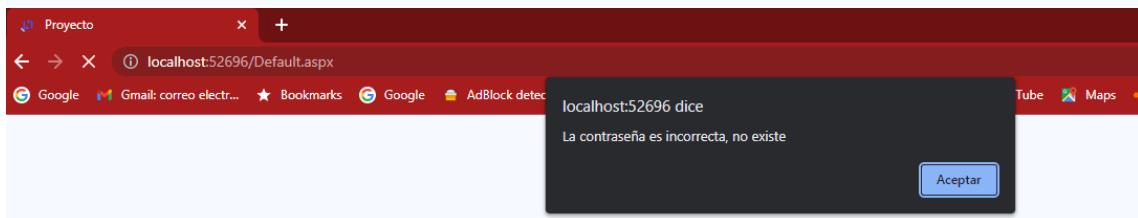
Usuario:

Contraseña:

Remember me

**Login**

Por vanyhony Esneyder Zabala Baez



Al iniciar el docente mostrará los datos del docente en la página de inicio, la imagen de perfil será esta por defecto si el docente no tiene foto de lo contrario se mostrará

Datos del docente			
Código Usuario:	3	Apellido 1:	Barreto
Código Docente:	75780748	Apellido 2:	Medina
Teléfono:	8	Nombre 1:	Zulma
Ingreso:	248583259	Nombre 2:	Rocío

**Actualización de docentes**

Código Usuario:	3	Apellido 1:	Zabala	Imagen:	<a href="#">ejemplo.png</a>
Código Docente:	114	Apellido 2:	Baez	Archivo:	
Teléfono:	3203590398	Nombre 1:	Vanyhony		
Ingreso	1516	Nombre 2:	Esneyder		

**Editar** **Eliminar**

The screenshot shows a web interface for managing teacher profiles. At the top, there are buttons for 'Actualizar' (Update) and 'Eliminar' (Delete). Below this is a table with teacher details. The right side of the screen displays a profile picture of a teacher and a quote by Joseph Joubert: "ENSEÑAR ES APRENDER DOS VECES".

Para cerrar sesión desplegar el perfil y seleccionar “sign out) se cerrará y volverá a la página principal:

The screenshot shows a dropdown menu from a user profile. The menu items are: My Profile, Account Settings, Need Help?, and Sign Out. The 'Sign Out' option is underlined, indicating it is selected.

### Explicación técnica para el logueo:

En el evento para loguearse, se genera una consulta teniendo en cuenta el código de usuario y el campo ingreso de la tabla docente, donde atreves de una variable booleana validara que exista el docente si no existe mostrara la alerta, y si existe guardara los datos sen variables de sesión teniendo en cuenta la clave, para así retornar el inicio de pagina para el docente, donde por un form view mostrara los datos del docente con la condición sql where del campo ingreso, si el usuario tiene imagen mostrara la foto guardada en el servidor con el nombre que tenga de imagen de lo contrario mostrara una imagen que está ahí por defecto en la carpeta.

Para evitar que se ingrese a la pagina de inicio del docente sin haber ingresado una contraseña se verifica las variables de sesión, que exista si no existe lo devolverá a la pagina principal, y al cerrar sesión estas variables de sesión se va a vaciar.

### Evento iniciar sesión:

```
protected void Button1_Click(object sender, EventArgs e)
{
    conexion.Open();
    if (DropDownList1.SelectedValue.Equals("3"))
    {
        if (TextBox1.Text.Equals(""))
        {
            Response.Redirect("Default.aspx");
        }
        else
        {
            string cadena = "select usucodigo, doccodigo, docapellido1, docapellido2, docnombre1, docnombre2, docingreso, imagen from tbldocentes";
            SqlCommand comando = new SqlCommand(cadena, conexion);
            SqlDataReader registros = comando.ExecuteReader();
            while (registros.Read())
            {
                if (registros["docingreso"].ToString().Equals(TextBox1.Text))
                {
                    apellido1 = (registros["docapellido1"].ToString());
                    apellido2 = (registros["docapellido2"].ToString());
                    nombre1 = (registros["docnombre1"].ToString());
                    nombre2 = (registros["docnombre2"].ToString());
                    imagen = (registros["imagen"].ToString());
                    codigo = (registros["doccodigo"].ToString());
                    us = $"{nombre1} {nombre2} {apellido1} {apellido2}";
                    usNA =($"{nombre1} {apellido1}");
                    condicion = true;
                }
            }
            if (condicion == false)
            {
                ScriptManager.RegisterStartupScript(this, this.GetType(),
                    "alert",
                    "alert('La contraseña es incorrecta, no existe');", true);
                conexion.Close();
                TextBox1.Text = "";
            }
            else
            {
                //Almacenamos las dos variables de sesión
                Session["codigo"] = codigo;
                Session["usuario"] = us;
                Session["usuarioNA"] = usNA;
                Session["imagen"] = imagen;
                Session["clave"] = TextBox1.Text;
                //Redireccionamos a la siguiente pagina
                Response.Redirect("Docinicio.aspx");
            }
        }
    }
}
```

### Evento de iniciar pagina para el inicio del docente:

```
public partial class Docinicio : System.Web.UI.Page
{
    0 referencias
    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.Session.Count == 0)
        {
            Response.Redirect("Default.aspx");
        }
        else
        {
            Label1.Text = $"Pagina de inicio Docente ({this.Session["usuario"].ToString()})";
        }
    }
}
```

Evento al cargar la pagina para el docente en el master Page:

```

1 referencia
public partial class MasterPageDoc : System.Web.UI.MasterPage
{
    String valorImagen;
0 referencias
protected void Page_Load(object sender, EventArgs e)
{
    valorImagen = this.Session["imagen"].ToString();
    MiLabel.Text = this.Session["usuario"].ToString();
    MiLabel1.Text = this.Session["usuario"].ToString();
    MiLabel2.Text = this.Session["usuarioNA"].ToString();
    if (valorImagen.Equals(""))
    {
        Image1.ImageUrl = "~/imagenes/imagenUsuario.png";
        Image2.ImageUrl = "~/imagenes/imagenUsuario.png";
    }
    else
    {
        Image1.ImageUrl = $"~/imagenes/{this.Session["imagen"].ToString()}";
        Image2.ImageUrl = $"~/imagenes/{this.Session["imagen"].ToString()}";
    }
}

```

Evento Botón de cerrar sesión:

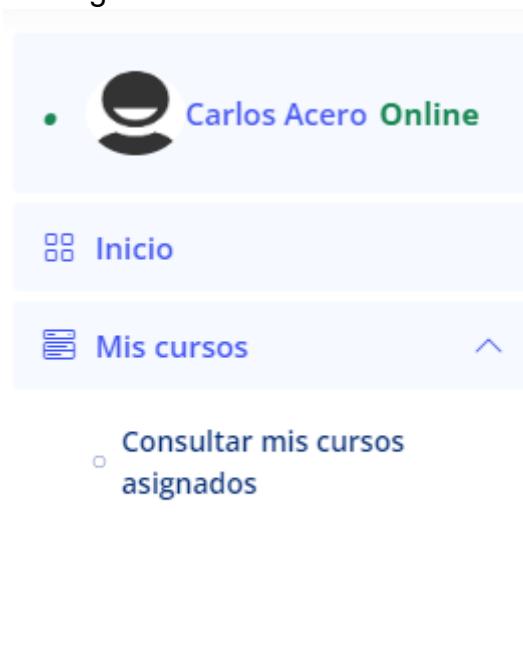
```

0 referencias
protected void Button1_Click1(object sender, EventArgs e)
{
    this.Session.Clear();
    Response.Redirect("Default.aspx");
}

```

### Consultar los cursos del docente:

En el menú estará los ítems de inicio y consultar cursos, donde podrá verse los cursos que tiene asignado el docente:



The screenshot shows a web interface for a teacher ('Docente'). At the top, there's a navigation bar with 'Docente' and a search bar. On the left, a sidebar has 'Carlos Acero Online' and 'Mis cursos'. The main content area displays a grid of courses with columns: 'Codigo del curso', 'Nombre del curso', 'Intensidad horaria', and 'Descripción del curso'. Two rows are visible: one for 'Ingles.' (code 1, intensity 5, 30 weeks) and another for 'Español.' (code 6, intensity 4, 20 weeks). Each row has a blue button labeled 'Ver estudiantes del curso'.

En la pestaña de ver los cursos como se evidencia se mostrara los cursos que tiene el docente, cada uno con un botón donde lo dirigirá a ver los estudiantes que tiene asignado a ese curso:

This screenshot shows a detailed view of the 'Ingles.' course from the previous screen. The title 'Curso Ingles.' is at the top. A gridview displays student information: 'Codigo del estudiante', 'Primer apellido', 'Segundo apellido', 'Primer nombre', and 'Segundo nombre'. Three students are listed: Barreto Esteban Julian, Garcia Avella Cyndy Jasbleidy, and Gonzalez Carmelo Rodrigo Benito. A green 'Volver' button is at the bottom left.

### Explicación técnica de los cursos de los docentes y sus estudiantes:

Para la ventana de consultar cursos se agrego un gridview apuntando a una vista creada a la base de datos, donde buscara los cursos por el código del docente ya declarado en una variable de sesión, en el botón se consultar estudiantes por curso entrara a la fila del gridview y capturara el código del curso y el código del docente, y los guarda en unas variables de sesión, Y redirige a la pagina donde buscara a los estudiantes inscritos por el curso y el docente logueado.

Código de los docentes del curso:

```
0 referencias
public partial class DcCursosDocente : System.Web.UI.Page
{
    0 referencias
    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.Session.Count == 0)
        {
            Response.Redirect("Default.aspx");
        }
    }

    0 referencias
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }

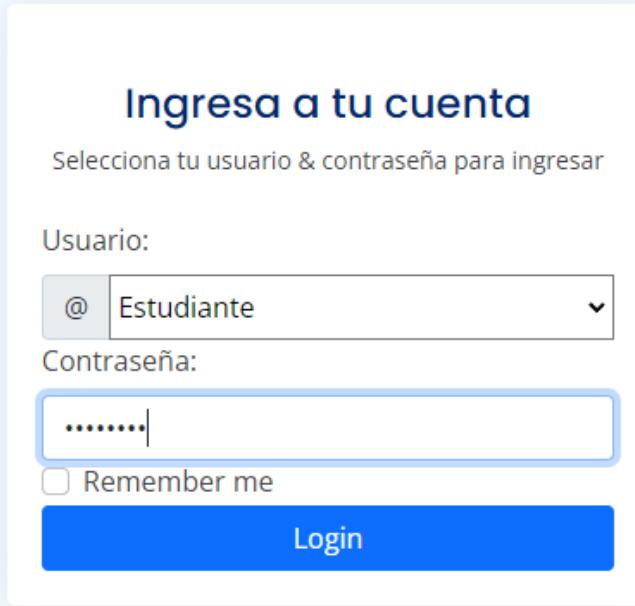
    0 referencias
    protected void GridView1_SelectedIndexChanged1(object sender, EventArgs e)
    {
        String codigo = GridView1.SelectedRow.Cells[0].Text;
        Session["verEstudianteCurso"] = codigo;
        Session["verEstudianteCursoNombre"] = GridView1.SelectedRow.Cells[1].Text;
        Response.Redirect("DcCursosDocentesEstudiantes.aspx");
    }
}
```

Código consulta estudiantes:

```
0 referencias
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["verEstudianteCurso"] != null && Session["verEstudianteCursoNombre"] != null)
    {
        if (!Session["verEstudianteCurso"].Equals("") && !Session["verEstudianteCursoNombre"].Equals(""))
        {
            Label5.Text = $"Curso {Session["verEstudianteCursoNombre"]}";
            Session["verEstudianteCursoNombre"] = "";
        }
        else
        {
            Response.Redirect("DcCursosDocente.aspx");
        }
    }
    else
    {
        Response.Redirect("DcCursosDocente.aspx");
    }
}
```

### Ingreso del estudiante:

Para el ingreso del estudiante, es elegir el usuario estudiante, y la contraseña será del campo ingreso con el que se registró.



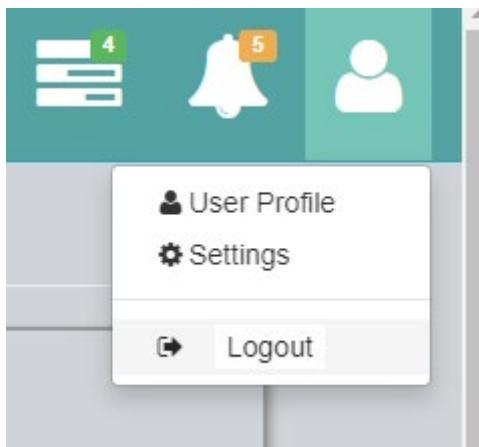
The image shows the login interface for the 'Proyecto ASP' system. At the top, there is a logo consisting of a blue 'L' and 'T' followed by the text 'Proyecto ASP'. Below the logo, the heading 'Ingresa a tu cuenta' (Log in to your account) is displayed in blue. A sub-instruction 'Selecciona tu usuario & contraseña para ingresar' (Select your user & password to log in) is shown in grey. The form fields include a dropdown menu for 'Usuario' (User) set to 'Estudiante', a password field containing '.....', and a 'Remember me' checkbox. A large blue 'Login' button is at the bottom. Below the login form, the text 'Por vanyhony Esneyder Zabala Baez' is displayed.

### Página de inicio del estudiante:



The image shows the student dashboard. At the top, there is a green header bar with the title 'ESTUDIANTE' and a user profile for 'maria jose gutierrez' (Online). Below the header, there is a sidebar with links for 'Inicio' and 'Cursos'. The main content area displays a profile card for 'Estudiante (maria jose gutierrez zabala)'. This card includes a 'Datos del Estudiante' section with details like 'Apellido 1: gutierrez', 'Apellido 2: zabala', 'Nombre 1: maria', and 'Nombre 2: jose'. Below this, there is a decorative illustration of a hand holding a stack of books, a diploma, a globe, and a trophy.

Para El estudiante será el mismo procedimiento que el de docentes, lo mismo para cerrar sesión:



En el menú se tendrá la opción cursos con ítems “consultar mi curso” donde el estudiante podrá ver los cursos que tiene asignado y con que docente:

The image shows a student dashboard with a green header bar. On the left, there's a sidebar with a user profile picture of 'maria gutierrez' (status 'Online'), a search bar, and navigation links for 'Inicio' and 'Cursos'. The main area has a title 'ESTUDIANTE' and a sub-section titled 'Estudiante (maria jose gutierrez zabala)'. Below this, a table displays course information:

	Código del curso	Nombre del curso	Intensidad horaria	Descripción	Código del docente	Docente
Seleccionar	8	Liderazgo.	9	Curso de formación corto 40 semanas	66983791	Avella Arcoz Doriz Estela

**Explicación técnica consulta cursos de estudiante:**

Se inserto un GridView con consulta de una vista Sql , con la condición where para la tabla tblcursoestudiantes, por la variable de sesión del código del estudiante, muestra los cursos a los que esta asignado , trayendo el curso y el docente que lo dicta.