

Grupo Alexandra De Gracia, Jaseth Castillo

Capturas de la interfaz.

Intefaz de admin/ profe



The image shows a screenshot of a web application window. The window's title bar is light blue and contains the text 'Registro' on the left, and standard window control buttons (minimize, maximize, close) on the right. The main content area has a light blue background. In the center, there is a white rectangular box with a thin black border. Inside this box, the title 'Acceso al sistema' is displayed in a large, black, serif font. Below the title, there are two input fields. The first is labeled 'Tipo de usuario' and is a dropdown menu with a white background and a small downward arrow on the right. The second is labeled 'Código de acceso' and is a text input field with a white background. At the bottom of the white box, there are two buttons: a green button with the text 'Ingresar' and a red button with the text 'Salir'.

Interfaz de registro de alumnos:

The screenshot shows a web application titled "Registro de Alumnos". It has a navigation bar with the following menu items: Nuevo, Guardar, Editar, Eliminar, Buscar, Salir, Acerca de, and Reportes. A dropdown menu for "Reportes" is open, showing "Reporte general".

The main content area is divided into two panels:

- Datos del Alumno:** A form for adding a new student with the following fields:
 - Id: 10
 - Nombre: (empty text input)
 - Cédula: (empty text input)
 - Carrera: (dropdown menu)
 - Semestre: (dropdown menu)
 - Radio buttons for "Vespertina" and "Matutina".
 - Checkboxes for "Acepto términos" and "Notificaciones".
- Lista de Alumnos:** A list of existing students with a search bar "Buscar por cédula" and a scrollable list of names and details:
 - 9 - Cesar Castillo - Ingeniería - Matutina
 - 8 - Are Montenegro - Ingeniería - Matutina
 - 7 - Sile - Medicina - Vespertina
 - 6 - Jaseth Castillo - Medicina - Vespertina

Below the "Datos del Alumno" panel is a "Confirmar:" section with three input fields for "Usuario", "Contraseña", and "Confirmar".

Código del CRUD completo.

INSERT (Crear Alumno) — GuardarAlumno()

```

2 referencias
private void GuardarAlumno()
{
    if (!ValidarCampos()) return;

    string jornada = rbMatutina.Checked ? "Matutina" : "Vespertina";

    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand(
        @"INSERT INTO Alumnos
        (Nombre, Cedula, Carrera, Semestre, Jornada, Usuario, Contraseña, RecibirNotificaciones)
        VALUES (@Nombre, @Cedula, @Carrera, @Semestre, @Jornada, @Usuario, @Contraseña, @Noti)", cn))
    {
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text.Trim());
        cmd.Parameters.AddWithValue("@Cedula", txtCedula.Text.Trim());
        cmd.Parameters.AddWithValue("@Carrera", cmbCarrera.Text);
        cmd.Parameters.AddWithValue("@Semestre", cmbSemestre.Text);
        cmd.Parameters.AddWithValue("@Jornada", jornada);
        cmd.Parameters.AddWithValue("@Usuario", txtUsuario.Text);
        cmd.Parameters.AddWithValue("@Contraseña", txtContraseña.Text);
        cmd.Parameters.AddWithValue("@Noti", chkNotificaciones.Checked);

        cn.Open();
        cmd.ExecuteNonQuery();
    }

    MessageBox.Show("Alumno guardado correctamente.", "Éxito");

    LimpiarCampos();
    CargarAlumnos();
    txtId.Text = ObtenerSiguienteId().ToString();
}

```

SELECT (Leer alumnos) — CargarAlumnos()

```

4 referencias
private void CargarAlumnos()
{
    lstAlumnos.Items.Clear();

    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand(
        "SELECT Id, Nombre, Carrera, Jornada FROM Alumnos ORDER BY FechaRegistro DESC", cn))
    {
        cn.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            int id = dr.GetInt32(0);
            string nombre = dr.GetString(1);
            string carrera = dr.GetString(2);
            string jornada = dr.GetString(3);

            lstAlumnos.Items.Add($"{id} - {nombre} - {carrera} - {jornada}");
        }
    }
}

```

SELECT por ID — CargarAlumnoPorId()

1 referencia

```
private void CargarAlumnoPorId(int id)
{
    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand("SELECT * FROM Alumnos WHERE Id = @Id", cn))
    {
        cmd.Parameters.AddWithValue("@Id", id);
        cn.Open();

        SqlDataReader dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            txtId.Text = dr["Id"].ToString();
            txtNombre.Text = dr["Nombre"].ToString();
            txtCedula.Text = dr["Cedula"].ToString();
            cmbCarrera.Text = dr["Carrera"].ToString();
            cmbSemestre.Text = dr["Semestre"].ToString();

            string jornada = dr["Jornada"].ToString();
            rbMatutina.Checked = jornada == "Matutina";
            rbVespertina.Checked = jornada == "Vespertina";

            txtUsuario.Text = dr["Usuario"].ToString();
            txtContraseña.Text = dr["Contraseña"].ToString();
            txtConfirmar.Text = dr["Confirmar"].ToString();
            chkNotificaciones.Checked = (bool)dr["RecibirNotificaciones"];
        }
    }
}
```

UPDATE — EditarAlumno()

```

private void EditarAlumno()
{
    if (string.IsNullOrEmpty(txtId.Text))
    {
        MessageBox.Show("Seleccione un alumno de la lista para editar.", "Advertencia");
        return;
    }

    if (!ValidarCampos()) return;

    string jornada = rbMatutina.Checked ? "Matutina" : "Vespertina";

    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand(
        @"UPDATE Alumnos SET
        Nombre = @Nombre,
        Cedula = @Cedula,
        Carrera = @Carrera,
        Semestre = @Semestre,
        Jornada = @Jornada,
        Usuario = @Usuario,
        Contraseña = @Contraseña,
        RecibirNotificaciones = @Noti
        WHERE Id = @Id", cn))
    {
        cmd.Parameters.AddWithValue("@Id", int.Parse(txtId.Text));
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text.Trim());
        cmd.Parameters.AddWithValue("@Cedula", txtCedula.Text.Trim());
        cmd.Parameters.AddWithValue("@Carrera", cmbCarrera.Text);
        cmd.Parameters.AddWithValue("@Semestre", cmbSemestre.Text);
        cmd.Parameters.AddWithValue("@Jornada", jornada);
        cmd.Parameters.AddWithValue("@Usuario", txtUsuario.Text);
        cmd.Parameters.AddWithValue("@Contraseña", txtContraseña.Text);
        cmd.Parameters.AddWithValue("@Noti", chkNotificaciones.Checked);

        cn.Open();
        cmd.ExecuteNonQuery();
    }

    MessageBox.Show("Alumno actualizado correctamente.", "Información");
    CargarAlumnos();
}

```

DELETE — EliminarAlumno()

2 referencias

```
private void EliminarAlumno()
{
    if (string.IsNullOrEmpty(txtId.Text))
    {
        MessageBox.Show("Seleccione un alumno de la lista para eliminar.", "Advertencia");
        return;
    }

    var confirmar = MessageBox.Show(
        "¿Está seguro que desea eliminar este alumno?",
        "Confirmación",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning);

    if (confirmar != DialogResult.Yes)
        return;

    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand("DELETE FROM Alumnos WHERE Id = @Id", cn))
    {
        cmd.Parameters.AddWithValue("@Id", int.Parse(txtId.Text));
        cn.Open();
        cmd.ExecuteNonQuery();
    }

    MessageBox.Show("Alumno eliminado correctamente.", "Información");

    LimpiarCampos();
    CargarAlumnos();
    LimpiarCampos();
}
```

Búsqueda por Cédula — BuscarAlumno()

1 referencia

```
private void buscaToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtbusc.Text))
    {
        MessageBox.Show("Ingrese una cédula para buscar.", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand("SELECT * FROM Alumnos WHERE Cedula = @Cedula", cn))
    {
        cmd.Parameters.AddWithValue("@Cedula", txtbusc.Text.Trim());

        cn.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            // Llenar los campos
            txtId.Text = dr["Id"].ToString();
            txtNombre.Text = dr["Nombre"].ToString();
            txtCedula.Text = dr["Cedula"].ToString();
            cmbCarrera.Text = dr["Carrera"].ToString();
            cmbSemestre.Text = dr["Semestre"].ToString();
            txtUsuario.Text = dr["Usuario"].ToString();
            txtContraseña.Text = dr["Contraseña"].ToString();
            chkNotificaciones.Checked = Convert.ToBoolean(dr["RecibirNotificaciones"]);

            string jornada = dr["Jornada"].ToString();
            rbMatutina.Checked = jornada == "Matutina";
            rbVespertina.Checked = jornada == "Vespertina";

            MessageBox.Show("Alumno encontrado.", "Resultado", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("No se encontró ningún alumno con esa cédula.", "Sin resultados", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

OBTENER SIGUIENTE ID

```
3 referencias
private int ObtenerSiguienteId()
{
    using (SqlConnection cn = Conexion.GetConnection())
    using (SqlCommand cmd = new SqlCommand(
        "SELECT ISNULL(MAX(Id), 0) + 1 FROM Alumnos", cn))
    {
        cn.Open();
        return Convert.ToInt32(cmd.ExecuteScalar());
    }
}
```

Código de ADO.NET.

CÓDIGO ADO.NET (Conexión, INSERT, UPDATE, DELETE, SELECT)

```
1  using System.Configuration;
2  using System.Data.SqlClient;
3
4  7 referencias
5  public static class Conexion
6  {
7      7 referencias
8      public static SqlConnection GetConnection()
9      {
10         string cnx = ConfigurationManager.ConnectionStrings["cnx"].ConnectionString;
11         return new SqlConnection(cnx);
12     }
13 }
```

Y app.config para la conexión con sql server :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="cnx"
          connectionString="Data Source=HunterWesterly\JASETHC;
                           Initial Catalog=DBAlumnos;
                           Integrated Security=True" />
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>
```

Código de validaciones y atajos.

```

2 referencias
private bool ValidarCampos()
{
    if (string.IsNullOrEmpty(txtNombre.Text) ||
        string.IsNullOrEmpty(txtCedula.Text))
    {
        MessageBox.Show("Nombre y cédula son obligatorios.");
        return false;
    }

    if (cmbCarrera.SelectedIndex == -1 || string.IsNullOrEmpty(cmbCarrera.Text))
    {
        MessageBox.Show("Debe seleccionar una carrera.");
        return false;
    }

    if (cmbSemestre.SelectedIndex == -1 || string.IsNullOrEmpty(cmbSemestre.Text))
    {
        MessageBox.Show("Debe seleccionar un semestre.");
        return false;
    }

    if (!rbMatutina.Checked && !rbVespertina.Checked)
    {
        MessageBox.Show("Debe seleccionar una jornada.");
        return false;
    }

    if (!chkTerminos.Checked)
    {
        MessageBox.Show("Debe aceptar los términos.");
        return false;
    }

    if (txtContraseña.Text != txtConfirmar.Text)
    {
        if (txtContraseña.Text.Length < 4)
        {
            MessageBox.Show("La contraseña debe tener al menos 4 caracteres.");
            return false;
        }

        if (txtUsuario.Text.Length == 0)
        {
            MessageBox.Show("No se pudo generar el usuario.");
            return false;
        }

        return true;
    }
}

```

Validación de cédula solo números

```

1 referencia
private void txtCedula_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
        e.Handled = true;
}

```

Validación de nombre solo letras


```

1 referencia
private void txtNombre_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && e.KeyChar != ' ')
        e.Handled = true;
}

```

Validación al salir del campo (Validating)

```

1 referencia
private void txtCedula_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrEmpty(txtCedula.Text))
    {
        MessageBox.Show("La cédula es obligatoria.", "Advertencia");
        e.Cancel = true;
    }
    else
        GenerarUsuario();
}

```

ATAJOS DE TECLADO (Ctrl + S, Ctrl + E, Ctrl + D, Escape)

```

226 private void Form1_KeyDown(object sender, KeyEventArgs e)
227 {
228     // Guardar
229     if (e.Control && e.KeyCode == Keys.S)
230     {
231         GuardarAlumno();
232         e.SuppressKeyPress = true;
233     }
234
235     // Editar
236     if (e.Control && e.KeyCode == Keys.E)
237     {
238         EditarAlumno();
239         e.SuppressKeyPress = true;
240     }
241
242     // Eliminar
243     if (e.Control && e.KeyCode == Keys.D)
244     {
245         EliminarAlumno();
246         e.SuppressKeyPress = true;

```

```

    }

    // Nuevo
    if (e.KeyCode == Keys.Escape)
    {
        LimpiarCampos();
        e.SuppressKeyPress = true;
    }

    // Reportes
    if (e.Control && e.KeyCode == Keys.R)
    {
        BtnReporte.PerformClick();
        e.SuppressKeyPress = true;
    }
}

```

Los 5 reportes impresos o capturados.

Reporte general:

VistaReporte

1 de 1 100 % Buscar | Siguiente

Reporte generales

Fecha: 25/11/2025 20:48:20

Id	Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Contraseña	Recibir Notificaciones	Fecha Registro
6	Jaseth Castillo	4835966	Medicina	1er Semestre	Vespertina	j4835966	12345	False	24/11/2025 11:25:06
7	Sile	13984293	Medicina	2do Semestre	Vespertina	s13984293	asdf	True	24/11/2025 16:56:58
8	Are Montenegro	13241231	Ingeniería	2do Semestre	Matutina	a13241231	5678	True	24/11/2025 21:00:09
9	Cesar Castillo	7329013821	Ingeniería	3er Semestre	Matutina	c7329013821	1234	False	25/11/2025 19:20:39

Página 1 de 1

Reporte de carrera:

VistaReporteCarrera

Ingeniería

Generar Reporte

1 de 1

100 %

Buscar | Siguiente

Reporte por carrera

25/11/2025
20:48:49

Carrera seleccionada:
Ingeniería

Id	Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Recibir Notificaciones
8	Are Montenegro	13241231	Ingeniería	2do Semestre	Matutina	a13241231	True
9	Cesar Castillo	7329013821	Ingeniería	3er Semestre	Matutina	c7329013821	False

VistaReporteCarrera

Medicina

Generar Reporte

1 de 1

100 %

Buscar | Siguiente

Reporte por carrera

25/11/2025
20:49:05

Carrera seleccionada:
Medicina

Id	Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Recibir Notificaciones
6	Jaseth Castillo	4835966	Medicina	1er Semestre	Vespertina	j4835966	False
7	Sile	13984293	Medicina	2do Semestre	Vespertina	s13984293	True

Página 1 de 1

Reporte por:

ReporteJornada

1 de 1 100 % Buscar | Siguiente

Reporte por Jornada

25/11/2025
20:49:29

Reporte agrupado por jornada	Id	Nombre	Cedula	Carrera	Semestre	Usuario	Recibir Notificaciones	Fecha Registro
Matutina	8	Are Montenegro	13241231	Ingenieria	2do Semestre	a13241231	True	24/11/2025 21:00:09
	9	Cesar Castillo	7329013821	Ingenieria	3er Semestre	c7329013821	False	25/11/2025 19:20:39
Vespertina	6	Jaseth Castillo	4835966	Medicina	1er Semestre	j4835966	False	24/11/2025 11:25:06
	7	Sile	13984293	Medicina	2do Semestre	s13984293	True	24/11/2025 16:56:58

Reporte hasta un rango de fecha:

ReporteRango

lunes , 24 de noviembre
martes , 25 de noviembre

Generar Reporte

1 de 1 100 % Buscar | Siguiente

Reporte de alumnos registrados en un rango de fechas

25/11/2025
20:50:02

Desde:
24/11/2025
Hasta:
25/11/2025

Id	Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Recibir Notificaciones	Fecha Registro
6	Jaseth Castillo	4835966	Medicina	1er Semestre	Vespertina	j4835966	False	24/11/2025 11:25:06
7	Sile	13984293	Medicina	2do Semestre	Vespertina	s13984293	True	24/11/2025 16:56:58
8	Are Montenegro	13241231	Ingenieria	2do Semestre	Matutina	a13241231	True	24/11/2025 21:00:09

Reporte de alumno elegido:

ReporteAlumno

Alumnos: Are Montenegi ▾

Generar reporte

1 de 1 100 % Buscar | Siguiente

Reporte de alumnos seleccionado

25/11/2025
20:50:32

Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Recibir Notificaciones	Fecha Registro
Are Montenegro	13241231	Ingeniería	2do Semestre	Matutina	a13241231	True	24/11/2025 21:00:09

Página 1 de 1

Explicación del flujo del sistema.

En este caso comenzamos con el forms de registro de administrados o profesor, si no ingresa la contraseña sale un aviso que no ha puesto ningún código, si pone una contraseña errónea, saldrá un mensaje de que esta mal la contraseña. Después de que el admin o profesor ingrese al sistema el podrá registrar un alumno con su: nombre, cédula, carrera, semestre, si es vespertino o matutino, y por último si quiere notificaciones y acepta los términos de condiciones, en este forms se podrá, actualizar, registrar, eliminar y buscar por cédula a cada alumno, además que se generará un usuario aleatorio dependiendo de la inicial y la cedula del alumno y una contraseña del usuario. Por último, tenemos la sección de reportes la cual el usuario elegirá que reporte quiere ver, si el de general, rango de fecha, por alumno, jornada, etc. Con todo lo anterior sumarle la validación y controles de atajos que el profe nos recomendo

Conclusiones del estudiante.

Conclusión (Alexandra De Gracia)

Este proyecto permitió entender cómo funciona un sistema CRUD, desde la captura de datos en la interfaz hasta el envío de instrucciones a SQL Server para guardarlos, mostrarlos, modificarlos o eliminarlos. Se aprendió a conectar correctamente la aplicación con la base de datos, validar la información y actualizar la tabla de manera automática para ver los cambios realizados.

En general, este trabajo ayudó a fortalecer conocimiento del C#, SQL Server y el uso de Windows Forms, demostrando cómo se combinan estos elementos para crear una aplicación funcional y práctica.

Conclusión (Jaseth Castillo)

La realización de este proyecto permitió comprender de manera práctica cómo se integra un sistema completo utilizando C#, SQL Server y Windows Forms. El desarrollo del CRUD mostró la importancia de manejar correctamente la interacción entre la interfaz gráfica y la base de datos, así como la validación de datos para asegurar la integridad de la información registrada.

Además, la implementación de los cinco reportes reforzó el uso de herramientas de generación de informes y el manejo de parámetros, mostrando cómo presentar la información de forma ordenada y profesional.