

Grupo Alexandra De Gracia, Jaseth Castillo

Capturas de la interfaz.

Intefaz de admin/ profe



Interfaz de registro de alumnos:

Registro de Alumnos

Nuevo Guardar Editar Eliminar Buscar Salir Acerca de Reportes Reporte general

– Datos del Alumno –

Id: 10

Nombre:

Cédula:

Carrera:

Semestre:

Vespertina  Matutina

Acepto términos  Notificaciones

– Lista de Alumnos –

Buscar por cédula:

9 - Cesar Castillo – Ingeniería – Matutina  
8 - Are Montenegro – Ingeniería – Matutina  
7 - Sile – Medicina – Vespertina  
6 - Jaseth Castillo – Medicina – Vespertina

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, Contrasena, RecibirNotificaciones)
        VALUES (@Nombre, @Cedula, @Carrera, @Semestre, @Jornada, @Usuario, @Contrasena, @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("@contrasena", txtContrasena.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["Contrasena"].ToString();
            txtConfirmar.Text = dr["Contrasena"].ToString();
            chkNotificaciones.Checked = (bool)dr["RecibirNotificaciones"];
        }
    }
}
```

UPDATE — EditarAlumno()

```

private void EditarAlumno()
{
    if (string.IsNullOrWhiteSpace(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,
Contrasena = @Contrasena,
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("@Contrasena", txtContrasena.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.IsNullOrWhiteSpace(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.IsNullOrWhiteSpace(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["Contrasena"].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  ↴  ↴  }
```

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.IsNullOrWhiteSpace(txtNombre.Text) ||
        string.IsNullOrWhiteSpace(txtCedula.Text))
    {
        MessageBox.Show("Nombre y cédula son obligatorios.");
        return false;
    }

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

    if (cmbSemestre.SelectedIndex == -1 || string.IsNullOrWhiteSpace(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.IsNullOrWhiteSpace(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
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246

```

```

1 referencia
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    // Guardar
    if (e.Control && e.KeyCode == Keys.S)
    {
        GuardarAlumno();
        e.SuppressKeyPress = true;
    }

    // Editar
    if (e.Control && e.KeyCode == Keys.E)
    {
        EditarAlumno();
        e.SuppressKeyPress = true;
    }

    // Eliminar
    if (e.Control && e.KeyCode == Keys.D)
    {
        EliminarAlumno();
        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:

The screenshot shows a Microsoft Word document titled "Reporte generales". The title is bold and centered at the top. Below it, there is a timestamp: "Fecha: 25/11/2025 20:48:20". A table follows, displaying student information across five rows. The columns are labeled: Id, Nombre, Cedula, Carrera, Semestre, Jornada, Usuario, Contrasena, Recibir Notificaciones, and Fecha Registro. The data is as follows:

Id	Nombre	Cedula	Carrera	Semestre	Jornada	Usuario	Contrasena	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 | Buscar | Siguiente | 100 %

## Reporte por carrera

25/11/2025  
20:48:49

Carrera seleccionada:  
Ingeniería

ID	Nombre	Cédula	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 | Buscar | Siguiente | 100 %

## Reporte por carrera

25/11/2025  
20:49:05

Carrera seleccionada:  
Medicina

ID	Nombre	Cédula	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 | Buscar | Siguiente | 100 %

### 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	Ingeniería	2do Semestre	a13241231	True	24/11/2025 21:00:09
	9	Cesar Castillo	7329013821	Ingeniería	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 | Buscar | Siguiente | 100 %

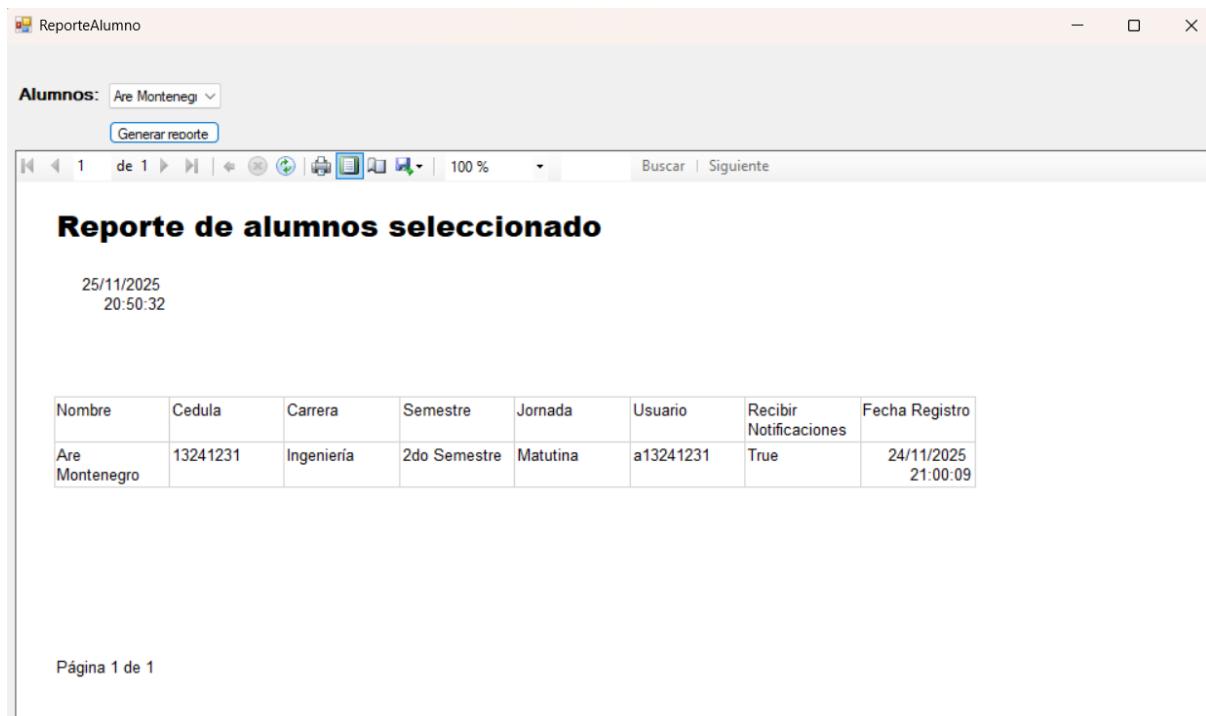
### Reporte de alumnos registrados en un rango de fechas

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	Ingeniería	2do Semestre	Matutina	a13241231	True	24/11/2025 21:00:09

Reporte de alumno elegido:



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 recomienda

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.