

Actividad | 3 | Creación de Interfaces (Lenguaje C#).

CRUD: Altas, Bajas, Modificaciones y Consultas.

Ingeniería en Desarrollo de
Software.



TUTOR: Miguel Ángel Rodríguez Vega.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 07/04/2024

Índice

Introducción	3
Descripción	3
Justificación	4
Desarrollo.....	4
Alta.....	5
Consulta	7
Modificación	13
Baja	16
Ejecución del programa	19
Conclusión	22
Referencias.....	23

Introducción

En la actividad final de la materia de lenguajes de programación III, nos plantea realizar la documentación necesaria para la creación de una programa en el lenguaje de programación C# utilizando la plataforma de programación de Microsoft Visual Studio 2022, Microsoft SQL Server donde nos permitirá la creación de un proyecto nuevo donde utilizaremos como base las opciones de la aplicación de escritorio con la compatibilidad de Windows Forms, C# y .NET Framework, estas herramienta será utilizadas para realizar la actividad en curso la cual no llevara a la calificación que se requiere para el acreditamiento de la materia de lenguajes de programación III lleva como nombre la actividad Creación de Interfaces utilizando el lenguaje de programación de C#.

Descripción

En esta actividad final entregaremos el documento del proyecto de nombre “CRUD alta, baja, modificación y consulta” utilizando la información de la actividad final proporcionada por el docente o maestro de la materia donde nos pide la creación de un programa sobre distintas actividades o acciones a realizar por parte el alumno de la universidad a interactuar con la aplicación a crear. Usando la herramienta de Microsoft Visual Studio 2022, Microsoft SQL Server y el lenguaje de programación de C#, en el cual requeriremos realizar 5 diferentes ventanas en donde se realizará lo siguiente: 1.-La pantalla principal 2.- Menú principal de la aplicación, con menú desplegable., 3.-Registros (Clientes, proveedores y productos)., 4.-Catalogo de Venta de la Zapatería UMI para damas y caballeros., 5.- Reporte de compras o ventas de la zapatería UMI, Todas estas ventanas conforman el proyecto a realizar para la actividad dos de la materia de Lenguajes de Programación III que se está cursando actualmente.

Justificación

En esta actividad trabajaremos con la documentación proporcionada por el docente de la materia de Lenguajes de Programación III se realizará la actividad de lenguaje C#, utilizando el método de programación con Windows Forms, .NET y C# en la herramienta de programación de Visual Studio, SQL Server donde tenemos que realizar lo solicitado por el docente:

- Utilizar la información con la que se ha trabajado en las actividades 2 entregada por el docente responsable de la materia.
- Subirlo al GitHub el documento realizado compartiendo el link para que pueda consultar el docente o maestro.
- Subir al GitHub los programas a realizar en la herramienta de visual studio 2022 al utilizar el lenguaje de programación de C#.
- Formato requerido por umi para la documentación de la actividad.
- Programa de escritorio con 5 ventana donde muestre los siguiente:
 - Primera ventana: Menú principal.
 - Segunda ventana: Registros (Cliente, Proveedor y Productos).
 - Tercera ventana: Catalogo de ventas de zapatería UMI.
 - Cuarta ventana: Reporte o consulta de compras en la zapatería UMI.
 - Quinta ventana: Mensajes de bienvenida, misio, visión.

Desarrollo:

En este punto de la actividad dos realizaremos el programa que se no solicita el cual tendrá 5 ventas diferentes donde mostrará la interfase y el diseño de cada una de sus partes al ser creado el proyecto con la herramienta de programación de Microsoft Visual Studio 2022, Microsoft SQL Server utilizando

Windows Forms con base de C# y .net framework en esta documentación anexaremos las imágenes de evidencia del programa realizado para este trabajo de la materia de lenguajes de programación III.

Link: [GitHub](#)

Alta.

En este punto de la actividad la evidencia de la realización de programa de escritorio utilizándola herramienta de programación Microsoft Visual Studio 2022, Microsoft SQL Server con la plantilla de Windows Forms y C#, anexaremos imágenes de evidencia de alta y el código que existe la ventana que realizará este resultado solicitado por el docente de la materia, en este programa trabajaremos en la información de una Zapatería de Nombre UMI.

ALTA: Evidencia de alta anexaremos como evidencia la pantalla del código se utiliza en registro de cliente, proveedor, productos que comparten el código en general.

```
private void btn_add_Click(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPII
    {
        // SqlCommand cmd = new SqlCommand("Insert into cliente(id_cliente, nombre, apellido, fec_nac, telefo
        // cmd.CommandType = CommandType.Text;
        cn.Open();
        if (string.IsNullOrEmpty(txtB_Nom.Text) ||
            string.IsNullOrEmpty(txtB_ape.Text) ||
            string.IsNullOrEmpty(txtB_fenac.Text) ||
            string.IsNullOrEmpty(txtB_pho.Text) ||
            string.IsNullOrEmpty(txtB_email.Text))
        {
            MessageBox.Show("Todos los campos son obligatorios para el alta del cliente.", "Operación incorrecta
        }
        else
        {
            using (SqlCommand cmd = new SqlCommand("INSERT INTO cliente (nombre, apellido, fec_nac, telefono, em
            {
                cmd.CommandType = CommandType.Text;

                // Add parameters to the command
                //cmd.Parameters.AddWithValue("@id_cliente", txtB_ID.Text);
                cmd.Parameters.AddWithValue("@nombre", txtB_Nom.Text);
                cmd.Parameters.AddWithValue("@apellido", txtB_ape.Text);
                cmd.Parameters.AddWithValue("@fec_nac", txtB_fenac.Text);
                cmd.Parameters.AddWithValue("@telefono", txtB_pho.Text);
                cmd.Parameters.AddWithValue("@email", txtB_email.Text);
                cmd.ExecuteNonQuery();

                txtB_Nom.Text = "";
                txtB_ape.Text = "";
                txtB_fenac.Text = "";
                txtB_pho.Text = "";
                txtB_email.Text = "";
                MessageBox.Show("Registro exitoso de cliente nuevo.", "Registro Cliente.");
            }
        }
    }
```

Registro Cliente: Zapateria UMI

Datos Personales.

ID:

Nombre(s):

Apellido(s):

Fecha Nacimiento:

Telefono:

E-Mail:

Agregar.

Editar.

Mostrar.

Salir.

Eliminar.

Registro Cliente.

Registro exitoso de cliente nuevo.

Aceptar

Registros

Proveedor Productos

Registro Proveedor

Nombre(s):

Apellido(s):

Direccion:

Telefono:

E-Mail:

ID:

El uso del compa ID es solamente para bajas o algun cambio de registro.

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

Registro Proveedor.

Registro exitoso de proveedor nuevo.

Aceptar

Registro Productos

Producto:

Marca:

Categoría:

Talla:

Costo:

Tipo:

Stock:

ID: El uso del campo ID se utiliza con los botones de baja y cambios.

Alta.

Cambio.

Ver.

Cerrar.

Baja.

Registro Proveedor.

Registro exitoso de producto nuevo.

Aceptar

	id_producto	nom_prod	Marc_prod	precio_prod	stock_prod	ctg	zise	tipc
	9	Botas de combate	Nine West	3500	50	Dama	27	Casi
	10	Botine	Nine West	1800	50	Dama	20	Casi
	11	Zapatos de vestir	Nine West	700	50	Dama	23	Casi
▶	12	Zapatos de tacón bajo	Nine West	1200	50	Dama	23	Casi
	13	Tenis	Adidas	1000	20	Caballero	27	Dep
	14	Oxford Lisos	Flexis	900	50	Caballero	28	Casi
	15	botines de vestir	Flexis	2000	50	Caballero	28	Casi

Consulta.

En este punto de la actividad anexaremos la evidencia de las consultas programa creado utilizando Windows Forms en la herramienta o plataforma de Visual Studio 2022, SQL Server donde tomaremos las pantallas de ejecución e interface como referencia para este punto de la actividad de la zapatería UMI solicitada por la materia de Lenguaje de Programación III.

Consulta: Evidencia de consulta anexaremos como evidencia la pantalla del código se utiliza en registro de cliente, proveedor y productos que comparten el código en general, la segunda consulta realizada fue en la pantalla de compras.


```
private void btt_vw_Click(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPII
    {
        SqlDataAdapter sda = new SqlDataAdapter("select * from cliente", cn);
        sda.SelectCommand.CommandType = CommandType.Text;
        cn.Open();
        sda.Fill(dt);
        dGV_db.DataSource = dt;
    }
}
```

Registro Cliente Zapateria UMI

Datos Personales.

ID:

Nombre(s):

Apellido(s):

Fecha Nacimiento:

Telefono:

E-Mail:

Agregar.

Editar.

Mostrar.

Salir.

Eliminar.

	id_cliente	nombre	apellido	fec_nac	telefono	email
▶	4	Mari Mar	Costeñita	02/02/1979	6677010203	marcost@gmail.c...
	5	Pablo	Escobar	12/01/1949	6672494949	pabito@gmail.com
*						

Registros
Proveedor Productos

Registro Proveedor

Nombre(s):

Apellido(s):

Direccion:

Telefono:

E-Mail:

ID: *El uso del compa ID es solamente para bajas o algun cambio de registro.*

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

	id_proveedor	nombre	apellido	telefono	email	direccion
▶	1	Pilar	Arco	6672930101	pila@gmail.com	Cumbres Borresco...
	2	Sonia	Camilo Gordillo	6674833845	scgtula@gmail.com	Tula #1111
•						

Registros

ProveedorProductos

Registro Productos

Producto:

Marca:

Categoria:

Talla:

Costo:

Tipo:

Stock:

ID:

Alta.

Cambio.

Ver.

Cerrar.

Baja.

El uso del campo ID se utiliza con los botones de baja y cambios.

	id_producto	nom_prod	Marc_prod	precio_prod	stock_prod	ctg	zise	tipi
▶	3	Tenis	Adidas	1000	20	Dama	25	Dep
	4	Mocasines	Flexi	1500	30	Dama	26	Cas
	5	Bailarinas	NINE WEST	1200	50	Dama	24	Cas
	6	Sandalias de tacon	Adidas	1000	20	Dama	23	Cas
	7	Mules	Flexis	900	50	Dama	21	Cas
	8	Zapatillas	Nine West	2000	50	Dama	22	Cas
<	9	Botas de combate	Nine West	3500	50	Dama	27	Cas

```
private void btt_vtas_Click(object sender, EventArgs e)
{
    string consulta = @"
    WITH MesesConVentas AS (
    SELECT DISTINCT YEAR(fch_comp) AS Año, MONTH(fch_comp) AS Mes
    FROM compras
    )

    SELECT Año, Mes,
    SUM(ctn_pod) AS Productos, FORMAT(SUM(cost_pod), 'C', 'es-MX') AS Compras
    FROM MesesConVentas
    LEFT JOIN compras ON MesesConVentas.Mes = MONTH(fch_comp) AND MesesConVentas.Año = YEAR(fch_comp)
    GROUP BY Año, Mes
    ORDER BY Año, Mes";

    using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPIII_SQLVS_A3;Persist Security Info=
    {
        try
        {
            cn.Open();
            SqlCommand command = new SqlCommand(consulta, cn);
            SqlDataReader reader = command.ExecuteReader();

            // Limpiar datos anteriores y definir columnas en el control DataGridView
            dataGridView1.Columns.Clear();
            dataGridView1.Columns.Add("Año", "Año");
            dataGridView1.Columns.Add("Mes", "Mes");
            dataGridView1.Columns.Add("Productos", "Productos");
            dataGridView1.Columns.Add("Compras", "Compras");

            // Limpiar filas anteriores
            dataGridView1.Rows.Clear();

            // Leer resultados y agregarlos al control DataGridView
            while (reader.Read())
            {
                // Agregar fila al DataGridView
                dataGridView1.Rows.Add(
                    reader["Año"].ToString(),
                    reader["Mes"].ToString(),
                    reader["Productos"].ToString(),
                    reader["Compras"].ToString()
                );
            }

            reader.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error al cargar las ventas: " + ex.Message);
        }
    }
}
```

Compras Zapateria UMI

Reporte Zapateria UMI

	Año	Mes	Productos	Compras
▶	2024	2	10	\$17.400.00
	2024	3	18	\$21.800.00
*	2024	4	7	\$7.500.00

Ventas Mes.

Salir.

```

1 referencia
private void rdoB_C_CheckedChanged(object sender, EventArgs e)
{
    if (rdoB_C.Checked)
    {
        string consulta = "SELECT DISTINCT nom_prod, CAST(zise AS INT) AS zise FROM producto WHERE ctg= 'Caballero' ORDER BY nom_prod ASC, zise ASC";

        using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPIII_SQLVS_A3;Persist Security Info=
        {
            cmbB_Prod.Items.Clear();
            cmbB_zse.Items.Clear();
            cmbB_clr.Items.Clear();

            HashSet<int> tamañosUnicos = new HashSet<int>(); // HashSet para almacenar tamaños únicos

            SqlCommand command = new SqlCommand(consulta, cn);

            try
            {
                cn.Open();
                SqlDataReader reader = command.ExecuteReader();
                while (reader.Read())
                {
                    cmbB_Prod.Items.Add(reader["nom_prod"].ToString());

                    int tamaño = Convert.ToInt32(reader["zise"]);
                    if (!tamañosUnicos.Contains(tamaño)) // Verificar si el tamaño ya existe en HashSet
                    {
                        cmbB_zse.Items.Add(tamaño.ToString());
                        tamañosUnicos.Add(tamaño); // Agregar el tamaño al HashSet
                    }
                }
                reader.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error al cargar los productos: " + ex.Message);
            }

            // Agregar colores al ComboBox de colores
            cmbB_clr.Items.Add("Blanco");
            cmbB_clr.Items.Add("Negro");
            cmbB_clr.Items.Add("Azul");
            cmbB_clr.Items.Add("Gris");
            cmbB_clr.Items.Add("Cafes");
        }
    }
}

```

```

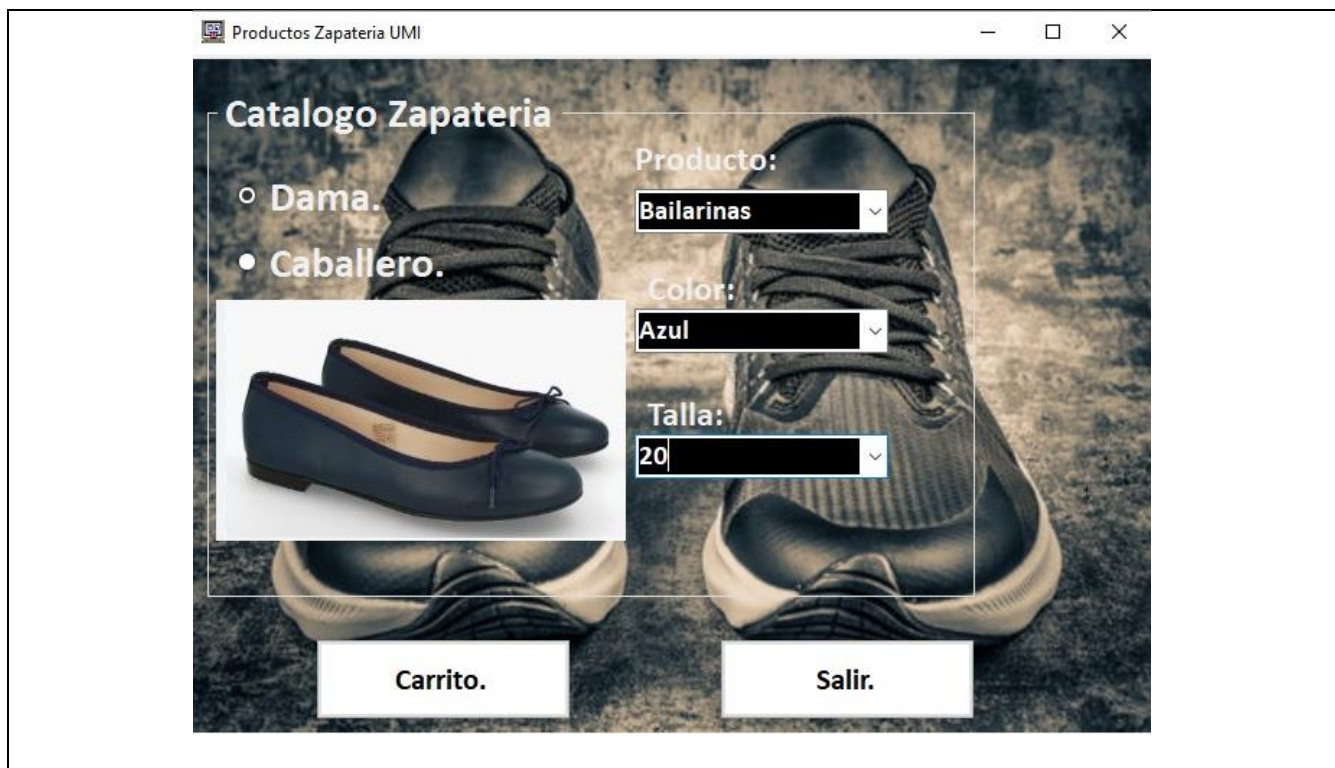
1 referencia
private void cmbB_Prod_SelectedIndexChanged(object sender, EventArgs e)
{
    string productoSeleccionado = cmbB_Prod.SelectedItem.ToString();
    string consultaImagen = "SELECT img FROM producto WHERE nom_prod = @producto";

    using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPIII_SQLVS_A3;Persist Security Info=
    using (SqlCommand command = new SqlCommand(consultaImagen, cn))
    {
        command.Parameters.AddWithValue("@producto", productoSeleccionado);

        try
        {
            cn.Open();
            byte[] imagenBytes = (byte[])command.ExecuteScalar(); // Obtener los bytes de la imagen directamente

            // Crear un MemoryStream a partir de los bytes de la imagen
            using (MemoryStream ms = new MemoryStream(imagenBytes))
            {
                // Cargar la imagen en el control PictureBox desde el MemoryStream
                picB_prod.Image = Image.FromStream(ms);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error al cargar la imagen del producto: " + ex.Message);
        }
    }
}

```



Modificación.

En este punto de la actividad anexaremos la evidencia de las modificaciones del programa creado utilizando Windows Forms en la herramienta o plataforma de Visual Studio 2022, SQL Server donde tomaremos las pantallas de ejecución e interface como referencia para este punto de la actividad de la zapatería UMI solicitada por la materia de Lenguaje de Programación III.

Modificación: Evidencia de la modificación anexaremos como evidencia la pantalla del código se utiliza en registro de cliente, proveedor y productos que comparten el código en general.


```

1 referencia
private void btt_edt_Click(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection("Data Source=231444L98188824\\SQLEXPRESS;Initial Catalog=DB_LPITI_SQLV5_A3;Persist Security Info=True;User ID=sa;Passw
    {
        cn.Open();

        string id = txtB_ID.Text;
        bool idExists = false;

        using (SqlCommand checkCmd = new SqlCommand("SELECT COUNT(*) FROM cliente WHERE id_cliente = @id", cn))
        {
            checkCmd.Parameters.AddWithValue("@id", id);
            int count = (int)checkCmd.ExecuteScalar();
            idExists = count > 0;
        }

        // Verificar si el ID existe y al menos un campo adicional no está vacío
        if (!idExists ||
            (string.IsNullOrEmpty(txtB_Nom.Text) &&
             string.IsNullOrEmpty(txtB_ape.Text) &&
             string.IsNullOrEmpty(txtB_fenac.Text) &&
             string.IsNullOrEmpty(txtB_pho.Text) &&
             string.IsNullOrEmpty(txtB_email.Text)))
        {
            MessageBox.Show("El ID ingresado no existe o no se proporcionó información adicional para actualizar el registro.", "Error de Validación");
        }
        else
        {
            // ID existe y al menos un campo adicional no está vacío, proceder con la actualización
            StringBuilder queryBuilder = new StringBuilder("UPDATE cliente SET ");

            if (!string.IsNullOrEmpty(txtB_Nom.Text))
            {
                queryBuilder.Append("nombre = @nombre, ");
            }
            if (!string.IsNullOrEmpty(txtB_ape.Text))
            {
                queryBuilder.Append("apellido = @apellido, ");
            }
            if (!string.IsNullOrEmpty(txtB_fenac.Text))
            {
                queryBuilder.Append("fec_nac = @fec_nac, ");
            }
            if (!string.IsNullOrEmpty(txtB_pho.Text))
            {
                queryBuilder.Append("telefono = @telefono, ");
            }
            if (!string.IsNullOrEmpty(txtB_email.Text))
            {
                queryBuilder.Append("email = @email, ");
            }

            queryBuilder.Length -= 2;
            queryBuilder.Append(" WHERE id_cliente = @id");

            using (SqlCommand cmd = new SqlCommand(queryBuilder.ToString(), cn))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@id", id);
            }
        }
    }
}

```

Registro Cliente Zapateria UMI

Registro Cliente

ID: Usar este campo en la
baja de un registro.

Nombre(s):

Apellido(s):

Fecha de Nacimiento:

Telefono:

E-Mail:

Alta.

Modificar.

Registro Cliente Modificado

El registro fue modificado de forma exitosa.

Aceptar

	id_cliente	nombre	apellido	fec_nac	telefono	email
	9	Adolfo	Hitler	20/04/1889	6673200489	AdolfH@gmail.com
▶	10	Cesar	Roma Imperial	12/03/1900	6677031200	cesamperio@g...
*						

Registros

Proveedor | Productos

Registro Proveedor

Nombre(s):

Apellido(s):

Direccion:

Telefono:

E-Mail:

ID:

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

El uso del compa ID es solamente pa

bajas o algun cambio de registro.

Registro Cliente Modificado

El registro fue modificado exitosa.

Aceptar

id_proveedor	nombre	apellido	telefono	email	direccion
1	Pilar	Arco	6672930101	plia@gmail.com	Cumbres Borresc...
2	Sonia	Camilo Gordillo	6674833845	scgtula@gmail.com	Tula #1111
3	Pablo	Pedro	6671987654	ppbro@gmail.com	Hermanos s/n
*					

Registro Productos

Producto:

Marca:

Categoría:

Talla:

Costo:

Tipo:

Stock:

ID: El uso del campo ID se utiliza con los botones de baja y cambios.

Alta.

Cambio.

Ver.

Cerrar.

Baja.

Registro Cliente Modificado

El registro fue modificado exitosamente.

Aceptar

	id_producto	nom_prod	Marc_prod	precio_prod	stock_prod	ctg	zise	tip
	3	Tenis	Adidas	1000	20	Dama	25	Dej
	4	Mocasines	Flexi	1500	30	Dama	26	Cas
	5	Bailarinas	NINE WEST	1200	50	Dama	24	Cas
	6	Sandalias de tacon	Adidas	1000	20	Dama	23	Cas
	7	Mules	Flexis	900	50	Dama	21	Cas
	8	Zapatillas	Nine West	2000	50	Dama	22	Cas
	9	Rotas de combate	Nine West	3500	50	Dama	27	Cas

Baja.

En este punto de la actividad anexaremos la evidencia de las modificaciones del programa creado utilizando Windows Forms en la herramienta o plataforma de Visual Studio 2022, SQL Server donde tomaremos las pantallas de ejecución e interface como referencia para este punto de la actividad de la zapatería UMI solicitada por la materia de Lenguaje de Programación III.

Baja: Evidencia de la baja anexaremos como evidencia la pantalla del código se utiliza en registro de cliente, proveedor y productos que comparten el código en general.

```
1 referencia
private void btt_elmn_Click(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection("Data Source=231444L90188024\\SQLEXPRESS;Initial Catalog=DB_LPITII_SQLVS_A3;Persist Security Info=True;User ID=sa;Password=1234567890"))
    {
        cn.Open();

        using (SqlCommand cmd = new SqlCommand("DELETE FROM cliente WHERE id_cliente = @id", cn))
        {
            cmd.Parameters.AddWithValue("@id", txtB_ID.Text);
            cmd.CommandType = CommandType.Text;

            int rowsAffected = cmd.ExecuteNonQuery();

            if (rowsAffected > 0)
            {
                MessageBox.Show($"Se eliminó el registro del cliente con el ID: {txtB_ID.Text}", "Éxito");
            }
            else
            {
                MessageBox.Show("No se encontró ningún cliente con el ID especificado, Usar el capo id.", "Aviso");
            }
        }
    }
}
```

Registro Cliente Zapateria UMI

Registro Cliente

ID:

Nombre(s):

Apellido(s):

Fecha de Nacimiento:

Telefono:

E-Mail:

Usar este campo en la baja de un registro.

Éxito

Se eliminó el registro del cliente con el ID: 11

Aceptar

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

	id_cliente	nombre	apellido	fec_nac	telefono	email
▶	9	Adolfo	Hitler	20/04/1889	6673200489	AdolfH@gmail.com
	10	Cesar	Roma	01/01/1900	6677010100	cesaroma@gmail...
	11	Rosario	Tijeras	12/12/1983	6672831212	rtijeras@gmail.com
•						

Registros

Proveedor | Productos

Registro Proveedor

Nombre(s):

Apellido(s):

Direccion:

Telefono:

E-Mail:

ID:

5

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

Éxito

Se eliminó el registro del Proveedor con el ID: 5

Aceptar

El uso del compa ID es solamente para bajas o algun cambio de registro.

	id_proveedor	nombre	apellido	telefono	email	direccion
▶	1	Pilar	Arco	6672930101	pilar@gmail.com	Cumbres Borresc...
	2	Sonia	Camilo Gordillo	6674833845	scgtula@gmail.com	Tula #1111
	3	Pablo	Pedro	6671987654	ppbro@gmail.com	Herradura #500
	5	Mojamed	Ali	6672345301	Ali2oro@gmail.com	Dos de oros
*						

The screenshot shows a Windows application window titled "Registros" with a tabbed interface. The "Productos" tab is active, displaying a "Registro Productos" form. The form has input fields for "Producto:", "Marca:", "Categoria:", "Talla:", "Costo:", "Tipo:", and "Stock:". The "ID:" field is populated with the value "23". To the right of the form are five buttons: "Alta.", "Cambio.", "Ver.", "Cerrar.", and "Baja.". A small dialog box titled "Éxito" is overlaid on the form, displaying the message "Se eliminó el registro del producto con el ID:" and an "Aceptar" button. Below the form is a table with the following data:

	id_producto	nom_prod	Marc_prod	precio_prod	stock_prod	ctg	zise	tipi
	19	Zapatos de taco...	Nine West	1200	50	Caballero	26	Cas
	20	Zapatos casual	Nine West	1800	50	Caballero	27	Cas
	21	Sandalias	Nine West	700	50	Caballero	28	Cas
	22	Zapatillas	Nine West	1200	50	Caballero	29	Cas
▶	23	Zapatos de taco...	Nine West	2100	30	Dama	24	Cas

Below the table, there is a text box with the message: "El uso del campo ID se utiliza con los botones de baja y cambios."

Ejecución del programa.

En este punto de la actividad anexaremos la evidencia de la ejecución completa de la aplicación creada utilizando Windows Forms en la herramienta o plataforma de Visual Studio 2022, SQL server tomando las pantallas de ejecución e interface como referencia para este punto de la actividad de cada ventana solicitada por la materia de Lenguaje de Programación III.

Ejecución: Mostraremos la evidencia de la ejecución y anexaremos la pantalla de la aplicación completa.



Registros
Proveedor Productos

Registro Proveedor

Nombre(s):

Apellido(s):

Dirección:

Teléfono:

E-Mail:

ID: *El uso del campo ID es solamente para bajas o algún cambio de registro.*

Alta.

Modificar.

Mostrar.

Salir.

Eliminar.

Operación incorrecta

Todos los campos son obligatorios para el alta del proveedor.

Aceptar

Registros
Proveedor Productos

Registro Productos

Producto:

Marca:

Categoría:

Talla:

Costo:

Tipo:

Stock:

ID: *El uso del campo ID se utiliza con los botones de baja y cambios.*

Alta.

Cambio.

Ver.

Cerrar.

Baja.



Conclusión

En conclusión: Las API CRUD son cruciales porque proporcionan una forma lógica y eficiente de crear, recuperar, actualizar y eliminar datos, lo cual es fundamental para administrar y escalar las operaciones de manera efectiva. La simplicidad y estandarización de las API CRUD agilizan los procesos de desarrollo e integración, reduciendo la complejidad y los costos.

Hablando de desarrollo de API, esta área está evolucionando y están surgiendo nuevas soluciones para simplificar aún más el proceso. Las soluciones de desarrollo de API sin código son cada vez más populares. Estas soluciones permiten a los usuarios diseñar e implementar API sin codificación, lo que hace que el desarrollo de API sea accesible a una audiencia más amplia, incluidos aquellos sin amplia experiencia técnica.

Referencias

GitHub: Let's build from here. (n.d.)

Ali, A. (2023, November 15). API CRUD: beneficios y técnicas de manejo de datos. *Astera*.

<https://www.astera.com/es/type/blog/crud-apis/>