



Actividad | 3 | Conexión de SQL

Server con Windows Forms

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 20 de Abril Del 2025.

Tabla de Contenido

Introducción	1
Descripción	1
Justificación	1
Desarrollo	2
Creación de la base de datos en “SQL Server”	2
Diseño y Codificación de Windows Forms	6
Conexión con la Base de Datos	27
Conclusión	55
Referencias	56

Introducción

Estamos ya en la última actividad de la materia de “Lenguajes de Programación III”, a lo largo de esta materia logramos trabajar con “Windows Forms” desde las cosas más básicas hasta más complejas y en esta actividad no será ninguna excepción puesto que trabajaremos en una base de datos en “SQL Server” para después diseñar nuestro formulario y con este lograr vincularlo para que todo lo que hagamos en “Windows Forms”, por ejemplo, agregar, eliminar o modificar se vea afectado directamente en nuestra tabla donde tenemos alojada la base de datos que se creó anteriormente, al igual que las otras actividades este documento tendrá el apartado de “Desarrollo” en donde se encontrara el progreso y ejecución de todo lo que se nos pide y al final en un link de “GitHub” estará el proyecto junto con toda la codificación para que el profesor pueda revisarlo sin ningún problema, al igual que se colocaran las referencias correspondientes de donde se tomaron toda la información.

Descripción

Empecemos con lo básico, en esta actividad veremos lo que es “SQL Server” y, aunque este término ya se vio en documentos pasados no está mal volver a conocerlo, pues bien “SQL Server” es un sistema de gestión de base de datos que es desarrollado por “Microsoft” es por ello que tiene una alta compatibilidad con “Visual Studio” y todas sus herramientas. Se utiliza para almacenar, gestionar y recuperar datos de manera eficiente, esto no afecta en su rendimiento si la información es pequeña o si se tiene una bitácora enorme con datos de todo el mundo, está diseñada que con relación al equipo de la empresa o persona que quiera alojar la información para que su lectura y escritura sea precisa y potente. La seguridad que maneja también es importante, una base de datos que aloja información sensible de personas, clientes, empleados o entre otros, es bastante imprescindible que exista una barrera antes de ingresar a esa información es por ello que “SQL Server” otorga una gran seguridad en donde los datos son encriptados y existen autenticaciones dentro de la aplicación mediante herramientas

Justificación

Ahora, ¿Por qué sería importante o necesario utilizar “SQL Server” ?, pues bien, esta herramienta nos permite gestionar y analizar datos de una manera eficiente en donde se acopla con las decisiones tomadas en organizaciones donde la información se guarda, se revisa o se modifica en tiempo real siempre manteniendo una capa de seguridad por encima de la información que se tiene en ella. La Eficiencia en la Gestión de Datos permite almacenar grandes cantidades de información, leerlas de manera rápida y precisa,

además de que también podemos realizar consultas en nuestra base de datos, un ejemplo simple sería el de revisar algún dato de una tabla en específico, bueno solo debes de colocar el comando con la sintaxis correcta, el nombre de la tabla que quieres ver y con eso será suficiente para que la ventana de información nos mande la ejecución del comando que escribimos y observemos lo que queremos ver.

Desarrollo

Creación de la base de datos en “SQL Server”

Para esto ingresaremos al software de “SQL Server” y crearemos una base de datos, como se muestra a continuación:

```
CREATE DATABASE Zapateria_la_Esperanza
```

Cuando ejecutamos el código con “F5” nos despliega un mensaje de que las filas se modificaron o se afectaron correctamente, es en ese instante que nuestra base de datos esta creada, refrescamos nuestra base de datos con el siguiente signo:



Esto sirve para actualizar los cambios que se hagan, después de eso nos fijamos en nuestra carpeta si se creó la base de datos:

Zapateria_la_Esperanza

Proseguimos a crear las tablas correspondientes, primero la tabla de clientes en donde tendrá la información de los mismos, para ello colocamos el siguiente código:

```
CREATE TABLE Cliente (
  DNI INT PRIMARY KEY IDENTITY,
  Nombre_Cliente VARCHAR (80) NOT NULL,
  Apellido_Pat VARCHAR (80) NOT NULL,
  Apellido_Mat VARCHAR (80) NOT NULL,
  Fecha_Nacimiento VARCHAR (80) NOT NULL,
  Telefono VARCHAR (80) NOT NULL,
)
```

Presionamos “F5” y nos dirigimos a actualizar la base de datos para saber si se creó correctamente nuestra tabla:

dbo.Cliente

Procedemos a colocar los datos de dicha tabla con el siguiente código:

```
INSERT INTO Cliente
(Nombre_Cliente, Apellido_Pat, Apellido_Mat, Fecha_Nacimiento, Telefono)
VALUES
('Sonia Alejandra', 'Fernandez', 'Moreno', '2003-06-16', '3366521589'),
('Daniela Fernanda', 'Diaz', 'Ordaz', '2000-02-25', '3354128951'),
('Mario Alberto', 'Jimenez', 'Salcido', '1995-07-01', '3321458971'),
('Yesenia Guadalupe', 'Campos', 'Rojo', '1998-12-12', '3354892655'),
('Tamara Alejandra', 'Bernal', 'Ramos', '2000-11-08', '3322548965'),
('Samuel Enrique', 'Barrios', 'Enciso', '2001-06-04', '3365217789'),
('Jorge Omar', 'Vazquez', 'Figueroa', '1999-10-29', '33214568844')
```

Nuevamente actualizamos y nos fijamos que los datos estén en la tabla correctamente:

	DNI	Nombre_Cliente	Apellido_Pat	Apellido_Mat	Fecha_Nacimiento	Telefono
1	1	Sonia Alejandra	Fernandez	Moreno	2003-06-16	3366521589
2	2	Daniela Fernanda	Diaz	Ordaz	2000-02-25	3354128951
3	3	Mario Alberto	Jimenez	Salcido	1995-07-01	3321458971
4	4	Yesenia Guadalupe	Campos	Rojo	1998-12-12	3354892655
5	5	Tamara Alejandra	Bernal	Ramos	2000-11-08	3322548965
6	6	Samuel Enrique	Barrios	Enciso	2001-06-04	3365217789
7	7	Jorge Omar	Vazquez	Figueroa	1999-10-29	33214568844



Seguimos ahora con la creación de la tabla de productos, para ello utilizaremos el siguiente código:

```
CREATE TABLE Producto (
Codigo INT PRIMARY KEY IDENTITY,
Género VARCHAR (80) NOT NULL,
Marca VARCHAR (80) NOT NULL,
Tipo VARCHAR (80) NOT NULL,
Talla VARCHAR (80) NOT NULL,
)
```

Continuamos con la colocación de datos de dicha tabla:

```
INSERT INTO Producto
(Género, Marca, Tipo, Talla)
VALUES
('Mujer', 'NIKE', 'Botas altas', '23'),
('Mujer', 'NIKE', 'Botas australianas', '23'),
('Mujer', 'PUMA', 'Botines', '23.5'),
('Mujer', 'PUMA', 'Merceditas', '24'),
('Mujer', 'CONVERSE', 'Zapatos de salón', '24.5'),
('Mujer', 'CONVERSE', 'Sandalias de Tacon', '25'),
('Mujer', 'PIRMA', 'Cuñas', '25.5'),
('Mujer', 'PIRMA', 'Mary Jane', '26'),
('Mujer', 'ADIDAS', 'Bailarinas', '26.5'),
('Mujer', 'ADIDAS', 'Sneakers', '27'),
('Hombre', 'NIKE', 'Entrenamiento', '27'),
('Hombre', 'NIKE', 'Crossfit', '27.5'),
('Hombre', 'PUMA', 'Caminata', '28'),
('Hombre', 'PUMA', 'Correr', '28.5'),
('Hombre', 'CONVERSE', 'Mocasines', '29'),
('Hombre', 'CONVERSE', 'Zapatos Náuticos', '29.5'),
('Hombre', 'PIRMA', 'Zapatos Monk', '30'),
('Hombre', 'PIRMA', 'Botas "Chelsea"', '30'),
('Hombre', 'ADIDAS', 'Zapatos "Brogue"', '30'),
('Hombre', 'ADIDAS', 'Sandalias', '30')
```

Al igual que las otras es importante actualizar y ver que se haya creado correctamente tanto la tabla como los datos:



  **dbo.Producto**

	Codigo	Género	Marca	Tipo	Talla
1	1	Mujer	NIKE	Botas altas	23
2	2	Mujer	NIKE	Botas australianas	23
3	3	Mujer	PUMA	Botines	23.5
4	4	Mujer	PUMA	Merceditas	24
5	5	Mujer	CONVERSE	Zapatos de salón	24.5
6	6	Mujer	CONVERSE	Sandalias de Tacon	25
7	7	Mujer	PIRMA	Cuñas	25.5
8	8	Mujer	PIRMA	Mary Jane	26
9	9	Mujer	ADIDAS	Bailarinas	26.5
10	10	Mujer	ADIDAS	Sneakers	27
11	11	Hombre	NIKE	Entrenamiento	27
12	12	Hombre	NIKE	Crossfit	27.5
13	13	Hombre	PUMA	Caminata	28
14	14	Hombre	PUMA	Correr	28.5
15	15	Hombre	CONVERSE	Mocasines	29
16	16	Hombre	CONVERSE	Zapatos Náuticos	29.5
17	17	Hombre	PIRMA	Zapatos Monk	30
18	18	Hombre	PIRMA	Botas "Chelsea"	30
19	19	Hombre	ADIDAS	Zapatos "Brogue"	30
20	20	Hombre	ADIDAS	Sandalias	30

Continuamos con la tabla de Proveedor para colocar el código:

```
CREATE TABLE Proveedor (
  NIF INT PRIMARY KEY IDENTITY,
  Nombre VARCHAR (80) NOT NULL,
  Dirección VARCHAR (80) NOT NULL,
  Correo VARCHAR (80) NOT NULL,
)
```

Revisamos que la tabla se creó:

  **dbo.Proveedor**

Insertamos los datos de la tabla:

```
INSERT INTO Proveedor
(Nombre, Dirección, Correo)
VALUES
('NIKE', 'LERDO DE TEJADA NO. 2628, ARCOS VALLARTA, 44130', 'nike@gmail.com'),
('PUMA', 'FCO I MADERO 310, CIUDAD REYNOSA CENTRO, 88500', 'puma@gmail.com'),
('PIRMA', 'REVOLUCION 2, BAHIA DE BANDERAS, 63732', 'pirma@gmail.com'),
('CONVERSE', 'REFORMA 304 NO. DEP 1 PLA BAJA, PUEBLA CENTRO, 72000', 'converse@gmail.com'),
('ADIDAS', 'PORFIRIO DIAZ NO. 473, CENTRO, 64000', 'adidas@gmail.com')
```


Revisamos que los datos en la tabla estén bien:

	NIF	Nombre	Dirección	Correo
1	1	NIKE	LERDO DE TEJADA NO. 2628, ARCOS VALLARTA, 44130	nike@gmail.com
2	2	PUMA	FCO I MADERO 310, CIUDAD REYNOSA CENTRO, 88500	puma@gmail.com
3	3	PIRMA	REVOLUCION 2, BAHIA DE BANDERAS, 63732	pima@gmail.com
4	4	CONVERSE	REFORMA 304 NO. DEP 1 PLA BAJA, PUEBLA CENTRO, 7...	converse@gmail.com
5	5	ADIDAS	PORFIRIO DIAZ NO. 473, CENTRO, 64000	adidas@gmail.com

Crearemos otra tabla de Compras, en donde se añadirá el precio de los productos, al igual que las anteriores se creará el código:

```
CREATE TABLE Compra (
Codigo INT PRIMARY KEY IDENTITY,
Género VARCHAR (80) NOT NULL,
Marca VARCHAR (80) NOT NULL,
Tipo VARCHAR (80) NOT NULL,
Talla VARCHAR (80) NOT NULL,
Precio FLOAT NOT NULL,
)
```

Procedemos a revisar que la tabla este correcta:

 **dbo.Compra**

Ingresamos los datos con el siguiente código:

```
INSERT INTO Compra
(Género, Marca, Tipo, Talla, Precio)
VALUES
('Mujer', 'NIKE', 'Botas altas', '23', '999.99'),
('Mujer', 'NIKE', 'Botas australianas', '23', '899.99'),
('Mujer', 'PUMA', 'Botines', '23.5', '599.99'),
('Mujer', 'PUMA', 'Merceditas', '24', '699.99'),
('Mujer', 'CONVERSE', 'Zapatos de salón', '24.5', '399.99'),
('Mujer', 'CONVERSE', 'Sandalias de Tacon', '25', '299.99'),
('Mujer', 'PIRMA', 'Cuñas', '25.5', '499.99'),
('Mujer', 'PIRMA', 'Mary Jane', '26', '559.99'),
('Mujer', 'ADIDAS', 'Bailarinas', '26.5', '259.99'),
('Mujer', 'ADIDAS', 'Sneakers', '27', '699.99'),
('Hombre', 'NIKE', 'Entrenamiento', '27', '899.99'),
('Hombre', 'NIKE', 'Crossfit', '27.5', '699.99'),
('Hombre', 'PUMA', 'Caminata', '28', '499.99'),
('Hombre', 'PUMA', 'Correr', '28.5', '799.99'),
('Hombre', 'CONVERSE', 'Mocasines', '29', '659.99'),
('Hombre', 'CONVERSE', 'Zapatos Náuticos', '29.5', '799.99'),
('Hombre', 'PIRMA', 'Zapatos Monk', '30', '899.99'),
('Hombre', 'PIRMA', 'Botas "Chelsea"', '30', '459.99'),
('Hombre', 'ADIDAS', 'Zapatos "Brogue"', '30', '559.99'),
('Hombre', 'ADIDAS', 'Sandalias', '30', '299.99')
```

Nuevamente revisamos que los datos estén correctos en la tabla:

	Codigo	Género	Marca	Tipo	Talla	Precio
1	1	Mujer	NIKE	Botas altas	23	999,99
2	2	Mujer	NIKE	Botas australianas	23	899,99
3	3	Mujer	PUMA	Botines	23.5	599,99
4	4	Mujer	PUMA	Merceditas	24	699,99
5	5	Mujer	CONVERSE	Zapatos de salón	24.5	399,99
6	6	Mujer	CONVERSE	Sandalias de Tacon	25	299,99
7	7	Mujer	PIRMA	Cuñas	25.5	499,99
8	8	Mujer	PIRMA	Mary Jane	26	559,99
9	9	Mujer	ADIDAS	Bailarinas	26.5	259,99
10	10	Mujer	ADIDAS	Sneakers	27	699,99
11	11	Hombre	NIKE	Entrenamiento	27	899,99

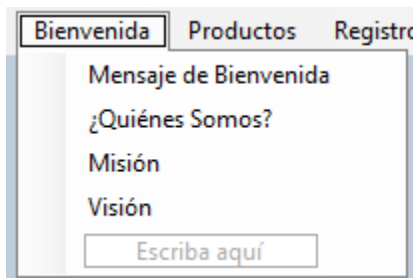
	Codigo	Género	Marca	Tipo	Talla	Precio
11	11	Hombre	NIKE	Entrenamiento	27	899,99
12	12	Hombre	NIKE	Crossfit	27.5	699,99
13	13	Hombre	PUMA	Caminata	28	499,99
14	14	Hombre	PUMA	Correr	28.5	799,99
15	15	Hombre	CONVERSE	Mocasines	29	659,99
16	16	Hombre	CONVERSE	Zapatos Náuticos	29.5	799,99
17	17	Hombre	PIRMA	Zapatos Monk	30	899,99
18	18	Hombre	PIRMA	Botas "Chelsea"	30	459,99
19	19	Hombre	ADIDAS	Zapatos "Brogue"	30	559,99
20	20	Hombre	ADIDAS	Sandalias	30	299,99

Diseño y Codificación de Windows Forms

Nos dirigiremos a “Windows Forms” para crear el diseño de nuestro menú principal, para esto como siempre empezamos con el título y con la imagen principal de la Universidad:



Procedemos a colocar un “MenuStrip” para el índice de nuestra aplicación, iniciando por “Bienvenida” y todos sus atributos:



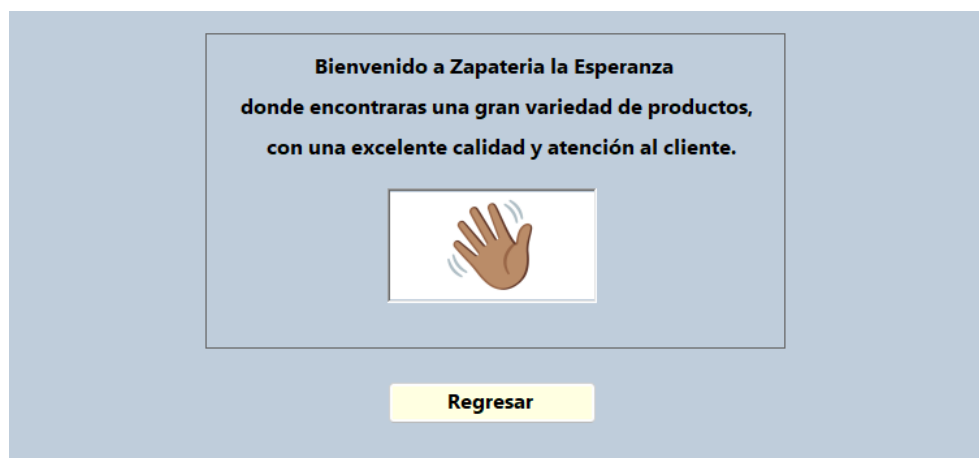
Terminaremos primero con la presentación del menú principal para ello agregaremos imágenes y demás cosas:



Procederemos a realizar el primer mensaje, lo haremos por medio de una nueva ventana, iniciamos por crear un nuevo formulario y creamos la parte superior:



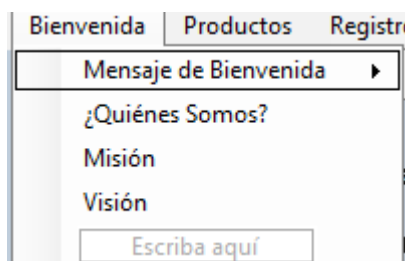
Y colocamos el mensaje con imágenes para que este más atractivo, tendrá un botón que nos regresara al menú principal:



Crearemos la codificación para el botón y que este nos regrese al inicio:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Tenemos que unir el formulario de mensaje de bienvenida con el de menú, para ello nos dirigimos al formulario principal y damos doble clic en “Mensaje de Bienvenida”:



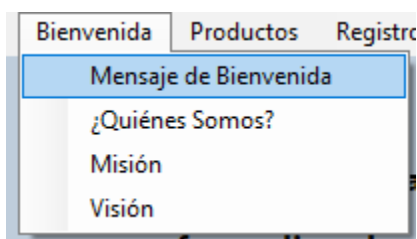
Y se coloca la codificación para que enlazar los formularios:

```
1 referencia
private void tool_Mensaje_Bienvenida_Click(object sender, EventArgs e)
{
    Formularios.Form_Mensaje_de_Bienvenida mensaje_De_Bienvenida= new Formularios.Form_Mensaje_de_Bienvenida();
    mensaje_De_Bienvenida.Show();
    this.Hide();
}
```

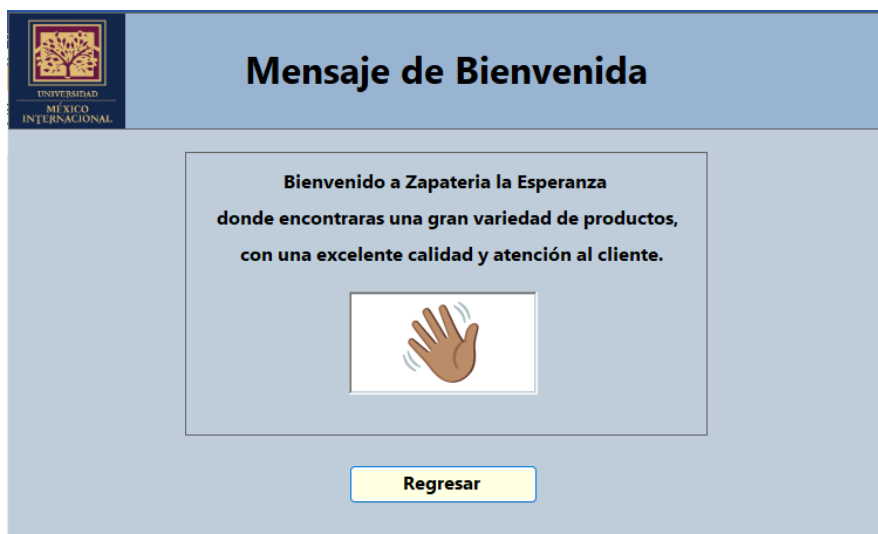
Iniciamos el programa y nos arroja el menú principal:



Seleccionamos “Mensaje de Bienvenida”:



Nos lleva al siguiente formulario sin ningún problema:



Probamos con el botón de “Regresar” y cómo podemos ver hace la función correctamente:



Crearemos la función de “Salir” para nuestro menú, damos doble clic en dicha función:



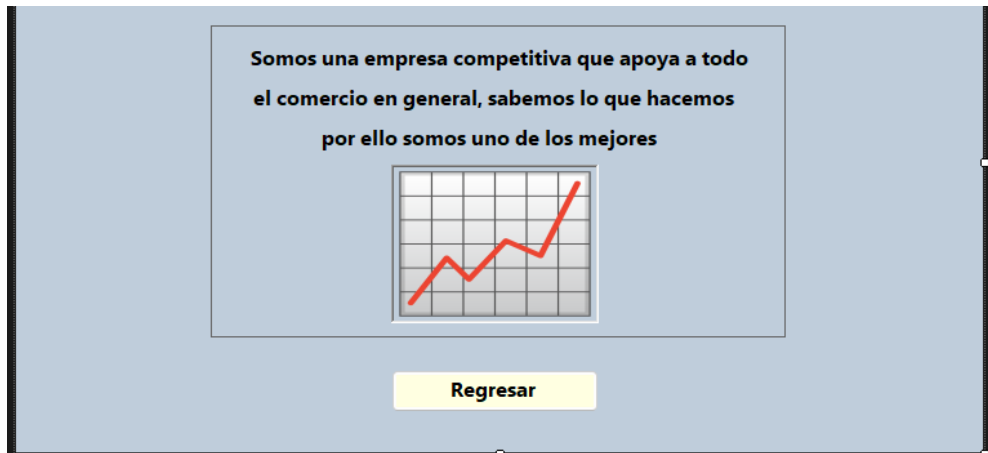
Colocamos el siguiente código, con esto es suficiente para que la aplicación se cierre por completo:

```
1 referencia
private void salirToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Haremos lo mismo con los otros formularios, el que sigue es “¿Quiénes Somos?”, primero crearemos un nuevo formulario y empezaremos con el diseño:



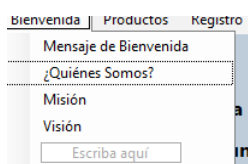
Continuamos con el cuerpo de nuestro formulario:



Igual que el anterior colocamos la función del botón de regresar:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Ahora procedemos a unir el formulario con el principal para esto damos doble clic en “¿Quiénes Somos?”:



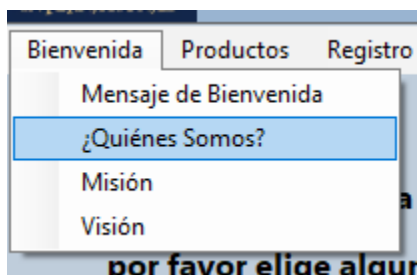
Colocamos el siguiente código:

```
1 referencia
private void tool_Quienes_Somos_Click(object sender, EventArgs e)
{
    Formularios.Form_Quienes_Somos form_Quienes_Somos = new Formularios.Form_Quienes_Somos();
    form_Quienes_Somos.Show();
    this.Hide();
}
```

Con esto nuestra ventana “¿Quiénes Somos?” esta enlazada, como siempre vamos a ejecutar el archivo para ver si todo funciona correctamente:



Seleccionamos el apartado “¿Quiénes Somos?” para ver si nos manda a la siguiente ventana:



En efecto, esto nos dirige a la ventana siguiente:



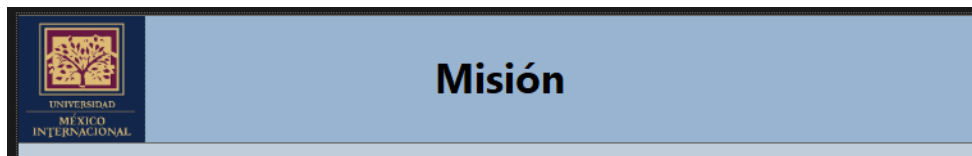
Presionamos el botón de “Regresar” y logramos ver que este nos manda al menú principal:



Si nosotros presionamos el botón de “Salir” que está en el menú nos cerrara la aplicación por completo:



Seguimos con la siguiente ventana de “Misión”, al igual que las otras nos iremos a crear el formulario y el diseño:



Seguimos con la parte del cuerpo:



Le damos la acción al botón de “Regresar”:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

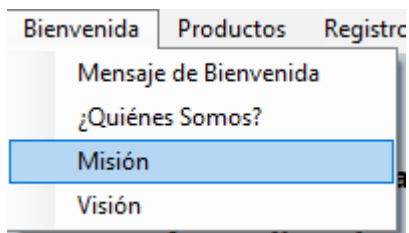
Y unimos el formulario con las mismas instrucciones que el anterior:

```
1 referencia
private void Tool_Misión_Click(object sender, EventArgs e)
{
    Formularios.Form_Mision form_Mision = new Formularios.Form_Mision();
    form_Mision.Show();
    this.Hide();
}
```

Ejecutamos el código para ver si funciona:



Damos clic en “Misión”:



Como podemos ver, nos manda a la otra ventana:



Utilizamos el botón de regresar para que nos mande al menú principal:



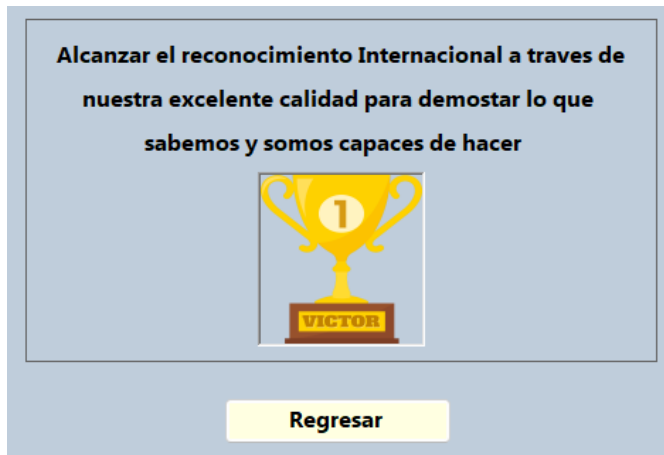
Para finalizar damos clic en "Salir":



Crearemos un último formulario en el que pondremos "Visión", para ello empezamos con el diseño:



Mostramos el cuerpo del diseño:



Colocamos la codificación en el botón de “Regresar”:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

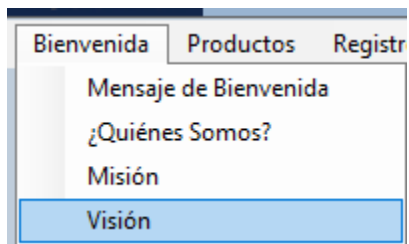
Ahora unimos las ventanas:

```
1 referencia
private void Tool_Vision_Click(object sender, EventArgs e)
{
    Formularios.Form_Vision form_Vision = new Formularios.Form_Vision();
    form_Vision.Show();
    this.Hide();
}
```

Ejecutamos el programa:



Seleccionamos “Visión”:



Observamos que nos manda a la siguiente ventana:



Regresamos al menú principal:



Y seleccionamos el botón de “Salir”:



Pasamos ahora con la sección de “Productos”, en ella se desplegarán dos opciones, una que diga “Damas” y otra que diga “Caballeros”, empezaremos primero con la de “Damas” para ello creamos un nuevo formulario y el diseño de la misma:

El cuerpo del formulario tiene la información de diez productos de calzado, con sus respectivas tallas en donde la “23” es la más chica y la “27” la más grande:

Marcas	Tipo	Tallas
NIKE	Botas Altas	<input type="checkbox"/> 23 <input type="checkbox"/> 23.5 <input type="checkbox"/> 24
NIKE	Botas Australianas	<input type="checkbox"/> 23 <input type="checkbox"/> 23.5 <input type="checkbox"/> 24
PUMA	Botines	<input type="checkbox"/> 23.5 <input type="checkbox"/> 24 <input type="checkbox"/> 24.5
PUMA	Merceditas	<input type="checkbox"/> 24 <input type="checkbox"/> 24.5 <input type="checkbox"/> 25
CONVERSE	Zapatos de Salón	<input type="checkbox"/> 24.5 <input type="checkbox"/> 25 <input type="checkbox"/> 25.5
CONVERSE	Sandalias de Tacón	<input type="checkbox"/> 25 <input type="checkbox"/> 25.5 <input type="checkbox"/> 26
PIRMA	Cuñas	<input type="checkbox"/> 25.5 <input type="checkbox"/> 26 <input type="checkbox"/> 26.5
PIRMA	Mary Jane	<input type="checkbox"/> 26 <input type="checkbox"/> 26.5 <input type="checkbox"/> 27
ADIDAS	Bailarinas	<input type="checkbox"/> 26.5 <input type="checkbox"/> 27
ADIDAS	Sneakers	<input type="checkbox"/> 27

Por supuesto este también tiene el botón de “Regresar” para que nos envíe al menú principal:

Nuevamente creamos la acción del botón de “Regresar”, damos doble clic para colocar el código:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Enlazamos las dos ventanas, para ello damos doble clic en “Damas”:

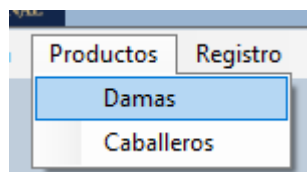
Escribimos el código de enlace:

```
1 referencia
private void Tool_Damas_Click(object sender, EventArgs e)
{
    Formularios.Form_Damas form_Damas = new Formularios.Form_Damas();
    form_Damas.Show();
    this.Hide();
}
```

Comprobamos que el código se ejecute correctamente:



Seleccionamos “Damas” en el menú:



Y como podemos observar es posible ver la ventana en donde nos arroja los productos que tenemos para “Damas”:

Marcas	Tipo	Tallas
NIKE	Botas Altas	<input type="checkbox"/> 23 <input type="checkbox"/> 23.5 <input type="checkbox"/> 24
NIKE	Botas Australianas	<input type="checkbox"/> 23 <input type="checkbox"/> 23.5 <input type="checkbox"/> 24
PUMA	Botines	<input type="checkbox"/> 23.5 <input type="checkbox"/> 24 <input type="checkbox"/> 24.5
PUMA	Merceditas	<input type="checkbox"/> 24 <input type="checkbox"/> 24.5 <input type="checkbox"/> 25
CONVERSE	Zapatos de Salón	<input type="checkbox"/> 24.5 <input type="checkbox"/> 25 <input type="checkbox"/> 25.5
CONVERSE	Sandalias de Tacón	<input type="checkbox"/> 25 <input type="checkbox"/> 25.5 <input type="checkbox"/> 26
PIRMA	Cuñas	<input type="checkbox"/> 25.5 <input type="checkbox"/> 26 <input type="checkbox"/> 26.5
PIRMA	Mary Jane	<input type="checkbox"/> 26 <input type="checkbox"/> 26.5 <input type="checkbox"/> 27
ADIDAS	Baillarinas	<input type="checkbox"/> 26.5 <input type="checkbox"/> 27
ADIDAS	Sneakers	<input type="checkbox"/> 27

Regresar

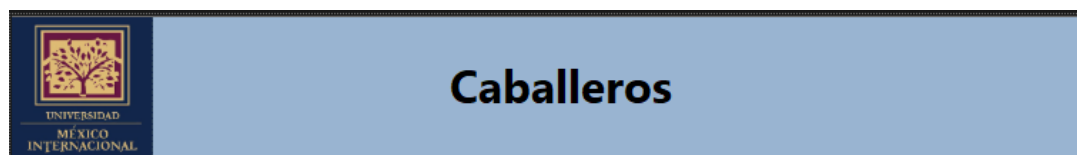
Damos clic en el botón de “Regresar y nos regresa al menú principal:



Ahora solamente damos clic en “Salir” para que nos cierre la aplicación:



Seguiremos con la sección de “Caballeros”, creamos un nuevo formulario y hacemos el diseño:



El cuerpo del Formulario esta con los diez productos y sus respectivas tallas, también está presente el botón de “Regresar”:

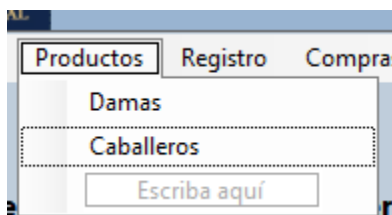
Marcas	Tipo	Tallas
NIKE	Entrenamiento	<input type="checkbox"/> 27 <input type="checkbox"/> 27.5 <input type="checkbox"/> 28
NIKE	Crossfit	<input type="checkbox"/> 27.5 <input type="checkbox"/> 28 <input type="checkbox"/> 28.5
PUMA	Caminata	<input type="checkbox"/> 28 <input type="checkbox"/> 28.5 <input type="checkbox"/> 29
PUMA	Correr	<input type="checkbox"/> 28.5 <input type="checkbox"/> 29 <input type="checkbox"/> 29.5
CONVERSE	Mocasines	<input type="checkbox"/> 29 <input type="checkbox"/> 29.5 <input type="checkbox"/> 30
CONVERSE	Zapatos Náuticos	<input type="checkbox"/> 29.5 <input type="checkbox"/> 30
PIRMA	Zapatos Monk	<input type="checkbox"/> 30
PIRMA	Botas "Chelsea"	<input type="checkbox"/> 30
ADIDAS	Zapatos "Brogue"	<input type="checkbox"/> 30
ADIDAS	Sandalias	<input type="checkbox"/> 30

Regresar

Creamos la codificación del botón de “Regresar”:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Damos doble clic en “Caballeros” que está en la parte del menú:



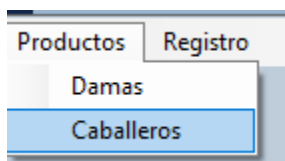
Colocamos la codificación para unir ambas ventanas:

```
1 referencia
private void caballerosToolStripMenuItem_Click(object sender, EventArgs e)
{
    Formularios.Form_Caballeros form_Caballeros = new Formularios.Form_Caballeros();
    form_Caballeros.Show();
    this.Hide();
}
```

Ejecutamos el código para revisar que todo esté funcionando correctamente:



Seleccionamos la sección de “Caballeros” que está en el menú principal:



Como podemos observar nos manda a la siguiente ventana en donde tenemos alojado los productos de “Caballeros”:

Marcas	Tipo	Tallas
NIKE	Entrenamiento	<input type="checkbox"/> 27 <input type="checkbox"/> 27.5 <input type="checkbox"/> 28
NIKE	Crossfit	<input type="checkbox"/> 27.5 <input type="checkbox"/> 28 <input type="checkbox"/> 28.5
PUMA	Caminata	<input type="checkbox"/> 28 <input type="checkbox"/> 28.5 <input type="checkbox"/> 29
PUMA	Correr	<input type="checkbox"/> 28.5 <input type="checkbox"/> 29 <input type="checkbox"/> 29.5
CONVERSE	Mocasines	<input type="checkbox"/> 29 <input type="checkbox"/> 29.5 <input type="checkbox"/> 30
CONVERSE	Zapatos Náuticos	<input type="checkbox"/> 29.5 <input type="checkbox"/> 30
PIRMA	Zapatos Monk	<input type="checkbox"/> 30
PIRMA	Botas "Chelsea"	<input type="checkbox"/> 30
ADIDAS	Zapatos "Brogue"	<input type="checkbox"/> 30
ADIDAS	Sandalias	<input type="checkbox"/> 30

Regresar

Seleccionamos el botón de “Regresar” y con ello nos manda de vuelta a la pantalla de inicio:

Zapateria la Esperanza

Bienvenida Productos Registro Compras Salir

Bienvenido a Zapateria la Esperanza
por favor elige alguna opción

😊

Ahora solo tenemos que dar clic en “Salir” para que cierre el programa por completo:

Salir

Nos dirigiremos al menú y colocaremos en “Registro” las siguientes opciones:

Registro Compras Salir

Cliente

Proveedor

Producto

Escriba aquí

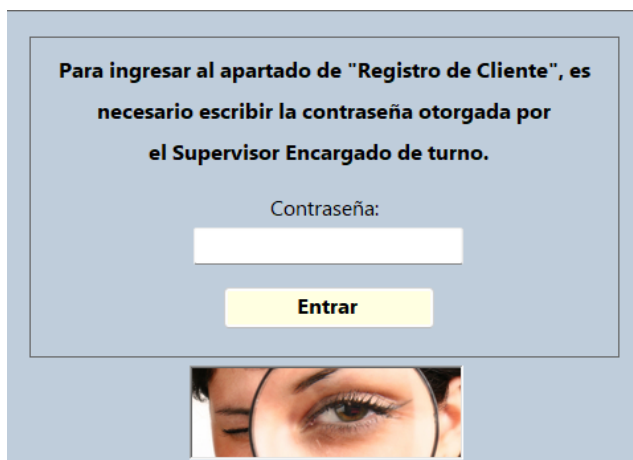
Trabajaremos primero con el de “Cliente”, pero antes de eso crearemos una ventana de ingreso que nos pedirá contraseña para proteger este formulario ya que no cualquiera tendrá acceso al mismo, vamos a crear un nuevo formulario y lo dejaremos en blanco:



Ahora crearemos otro formulario para hacer el diseño de “Ingreso” al registro de clientes:



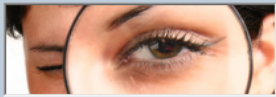
En el cuerpo del formulario se colocó un mensaje en el que deben ingresar la contraseña para acceder al siguiente formulario:



Para ingresar al apartado de "Registro de Cliente", es necesario escribir la contraseña otorgada por el Supervisor Encargado de turno.

Contraseña:

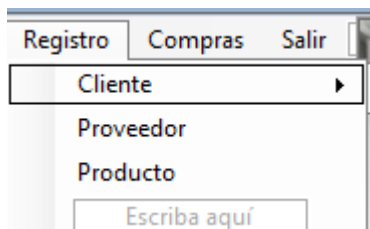
Entrar



Ahora bien, vamos a crear la codificación para hacer la acción de el botón “Entrar” y que nos dirija al formulario que queremos, en este caso “Registro de Clientes”, en este caso colocamos que si en el cuadro de texto de la contraseña no se escribe nada mandara un mensaje que “Favor de Ingresar la Contraseña”, si se coloca la contraseña correcta que es “963852741” direccionara en automático al siguiente formulario, ahora que si ninguna de las dos se cumple, pues nos mostrara un mensaje de que “La Contraseña es Incorrecta” y limpiara en automático el “TextBox”:

```
1 referencia
private void btn_Entrar_Cliente_Click(object sender, EventArgs e)
{
    if (txt_Contraseña_Cliente.Text == "")
    {
        MessageBox.Show("Favor de Ingresar la Contraseña.");
    }
    else if (txt_Contraseña_Cliente.Text == "963852741")
    {
        Form_Registro_Cliente _Registro_Cliente = new Form_Registro_Cliente();
        _Registro_Cliente.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("La Contraseña es Incorrecta.");
        txt_Contraseña_Cliente.Clear();
    }
}
```

Nos dirigimos al apartado de menú y en la sección de “Registro”, “Cliente” daremos doble clic para vincular el cuadro de ingreso con el menú principal:



Y escribimos la codificación correspondiente:

```
1 referencia
private void Tool_Cliente_Click(object sender, EventArgs e)
{
    Formularios.Form_Ingreso_Cliente form_Ingreso_Cliente = new Formularios.Form_Ingreso_Cliente();
    form_Ingreso_Cliente.Show();
    this.Hide();
}
```

Antes de ejecutar el código, tenemos el formulario en blanco que es el “Registro de Clientes” vamos a diseñarlo para darle ciertas acciones y después pasar a la conexión con la base de datos:

El formulario cuenta con un apartado en el que coloca los datos de acuerdo a los que vienen en la tabla de “SQL Server” al igual que botones de acción, faltaría colocar la codificación y la herramienta para mostrar la tabla, pero eso lo haremos después:

Vamos a colocarle una acción al botón de “Regresar” para que nos dirija al menú principal:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Por el momento crearemos otro formulario para el “Registro de Proveedor”, creamos un nuevo formulario con las mismas acciones que el anterior:

Le asignaremos la codificación al botón de “Regresar”:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Crearemos un formulario de “Ingreso” para acceder a “Registro de Proveedor”, como siempre empezaremos por el diseño:



Y en el cuerpo tenemos la misma información que el formulario de ingreso anterior excepto por el mensaje que esta vez dice “Registro de Proveedor”:



Creamos la misma codificación para que nos pida la contraseña, damos doble clic en el botón de “Entrar”:

```
1 referencia
private void btn_Enterar_Cliente_Click(object sender, EventArgs e)
{
    if (txt_Contraseña_Proveedor.Text == "")
    {
        MessageBox.Show("Favor de Ingresar la Contraseña. ");
    }
    else if (txt_Contraseña_Proveedor.Text == "963852741")
    {
        Form_Registro_Proveedor form_Registro_Proveedor = new Form_Registro_Proveedor();
        form_Registro_Proveedor.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("La Contraseña es Incorrecta. ");
        txt_Contraseña_Proveedor.Clear();
    }
}
```

Nos dirigimos al menú y seleccionamos “Proveedor” para enlazar las ventanas:

```
1 referencia
private void Tool_Proveedor_Click(object sender, EventArgs e)
{
    Formularios.Form_Ingreso_Registro_Proveedor form_Ingreso_Registro_ = new Formularios.Form_Ingreso_Registro_Proveedor();
    form_Ingreso_Registro_.Show();
    this.Hide();
}
```

Crearemos el Formulario de “Registro de Productos”, para ello empezamos por el diseño:

Le damos acción al botón de “Regresar”:

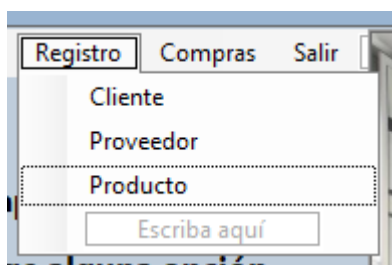
```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menú back = new Form_Menú();
    back.Show();
}
```

Creamos un nuevo formulario en el que estará el ingreso al anterior, para esto como siempre empezamos con el diseño:

Ahora creamos la codificación del botón de “Entrar”:

```
1 referencia
private void btn_Entrar_Cliente_Click(object sender, EventArgs e)
{
    if (txt_Contraseña_Producto.Text == "")
    {
        MessageBox.Show("Favor de Ingresar la Contraseña. ");
    }
    else if (txt_Contraseña_Producto.Text == "963852741")
    {
        Form_Registro_Producto form_Registro_Producto = new Form_Registro_Producto();
        form_Registro_Producto.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("La Contraseña es Incorrecta. ");
        txt_Contraseña_Producto.Clear();
    }
}
```

Nos vamos al apartado de menú y seleccionamos “Producto”, damos doble clic y colocamos el enlace:



Colocamos la codificación correspondiente:

```
1 referencia
private void Tool_Producto_Click(object sender, EventArgs e)
{
    Formularios.Form_Ingreso_Producto form_Ingreso_Producto = new Formularios.Form_Ingreso_Producto();
    form_Ingreso_Producto.Show();
    this.Hide();
}
```

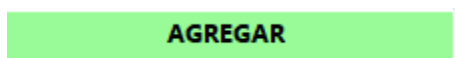
Conexión con la Base de Datos

Sigue lo más importante de esto, tenemos que enlazar la base de datos con el formulario de Windows Forms, damos clic en el formulario de “Registro de Clientes” y tenemos que agregar la siguiente librería:

```
using System.Data;
```

```
using System.Data.SqlClient;
```

Ahora seleccionamos el botón de “Agregar” que está en el “Registro de Clientes”:



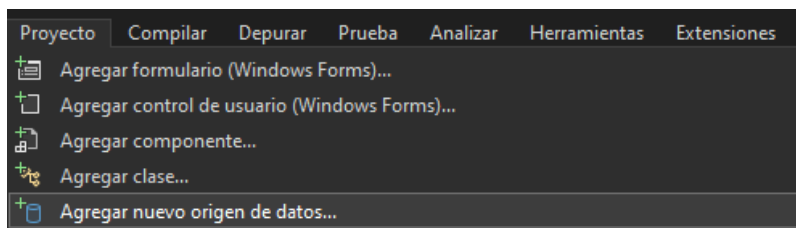
Nos abrirá la siguiente ventana:

```
1 referencia
private void btn_Agregar_Cliente_Click(object sender, EventArgs e)
{
    ...
}
```

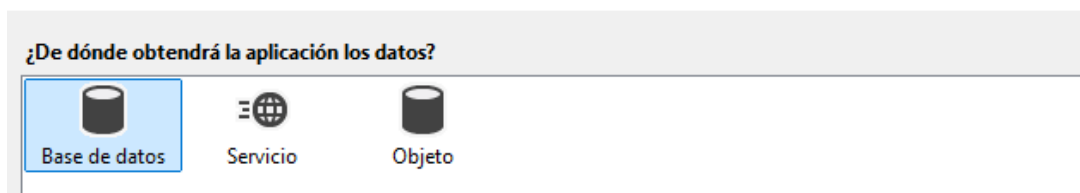
Colocamos “using” para después decirle que vamos a hacer una conexión con “SQL Server”, después mandamos a llamar la variable, se crea una nueva instancia y en donde nos quedamos tenemos que poner la dirección de la base de datos:

```
1 referencia
private void btn_Agregar_Cliente_Click(object sender, EventArgs e)
{
    using (SqlConnection Conexion = new SqlConnection("..."))
    {
        ...
    }
}
```

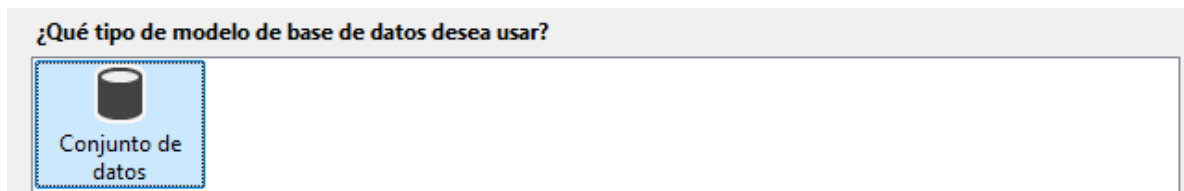
Para agregar la dirección tenemos que dirigirnos al apartado de “Proyecto” y después “Agregar un nuevo origen de datos”:



Nos abrirá la siguiente ventana en donde seleccionaremos “Base de datos”:



Se nos mostrara otra ventana, igual le damos en “Siguiente”:



Y nos mandara a otra ventana en donde tenemos varias características:

Seleccionamos “Nueva Conexión” y la opción de “Microsoft SQL Server” es la que seleccionaremos:

Nos abrirá una ventana aún más grande:

En este apartado tenemos que copiar la dirección que nosotros tenemos de nuestro servidor de “SQL Server”:

Nombro del servidor:

Para ello nos vamos a “SQL Server” y nos volvemos a conectar para que nos arroje una ventana en donde hay diferentes datos:

Esto de aquí lo copiamos para ingresarlo al otro apartado:

Quedaría de la siguiente manera:

Nombro del servidor:

En el siguiente apartado cambiamos la opción a como se muestra en la imagen para que nos deje seleccionar nuestras bases de datos que tenemos alojada:

Cifrar:

Seleccionamos la base de datos que tenemos en “SQL Server”:

☒ Seleccionar o escribir el nombre de la base de datos:

Damos clic en “Aceptar”:

Y nos manda de vuelta a la ventana principal en donde tendremos que darle clic en el siguiente apartado:

☒ Mostrar la cadena de conexión que se guardará en la aplicación

Y nos otorgara la conexión de la base de datos como se muestra abajo, eso lo tenemos que copiar:

```
Data Source=ROG-ONE\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;Integrated Security=True;Encrypt=False
```

Colocamos la dirección en el apartado que dejamos solo en “Visual Studio”:

```
SqlConnection("Data Source=ROG-ONE\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;Integrated Security=True;Encrypt=False")) ;|
```

Ponemos el código para que el comando lo guarde en la variable:

```
SqlCommand cmd = new SqlCommand("INSERT INTO Cliente (Nombre_Cliente, Apellido_Pat, Apellido_Mat, Fecha_Nacimiento, Te  
Fecha_Nacimiento, Telefono) VALUES ('" + txt_Nombre_Cliente.Text + "', '" + txt_ApellidoPat.Text + "', '" + txt_ApellidoMat.Text + "',  
doPat.Text + "', '" + txt_ApellidoMat.Text + "', '" + txt_Fecha_Nac.Text + "', '" + txt_Telefono.Text + "')", Conexion);
```

Ingresamos los comandos correspondientes para que la base de datos la lea:

```
cmd.CommandType = CommandType.Text  
Conexion.Open();  
cmd.ExecuteNonQuery();
```

Para limpiar todo cuando ingresemos los datos colocamos lo siguiente:

```
txt_Nombre_Cliente.Clear();  
txt_ApellidoPat.Clear();  
txt_ApellidoMat.Clear();  
txt_Fecha_Nac.Clear();  
txt_Telefono.Clear();
```

Y con un mensaje de que el cliente se agregó correctamente:

```
MessageBox.Show("El Cliente ha sido Agregado Correctamente. ");
```

Hacemos lo mismo con el botón de “Modificar”:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza
```

Código para que lo guarde la variable:

```
SqlCommand cmd = new SqlCommand("UPDATE Cliente SET Apellido_Pat = '"+ txt_ApellidoPat.Text + "', Apellido_Mat = '"+ tx
doPat.Text + "', Apellido_Mat = '"+ txt_ApellidoMat.Text + "', Fecha_Nacimiento = '"+ txt_Fecha_Nac.Text + "', Telefono = '"+ txt_Telefono
'+ txt_Telefono.Text + "' Where Nombre_Cliente = '"+ txt_Nombre_Cliente.Text + "' ", Conexion);
```

El comando para que lo ejecute:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

Para limpiar las “TextBox”;

```
txt_Nombre_Cliente.Clear();
txt_ApellidoPat.Clear();
txt_ApellidoMat.Clear();
txt_Fecha_Nac.Clear();
txt_Telefono.Clear();
```

Y el mensaje de que “El Cliente se ha Modificado Correctamente”:

```
MessageBox.Show("El Cliente ha sido Modificado Correctamente. ");
```

Igual con el botón de “Eliminar”:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;Integ
,
```

Código:

```
SqlCommand cmd = new SqlCommand("DELETE FROM Cliente WHERE Nombre_Cliente = '"+ txt_Nombre_Cliente.Text + "'", Conexion);
```

La ejecución del comando:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

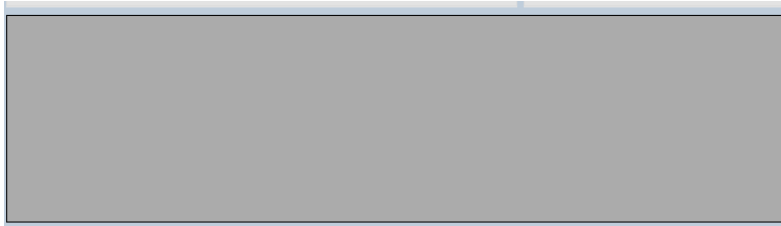
Para limpiar las “TextBox”:

```
txt_Nombre_Cliente.Clear();
txt_ApellidoPat.Clear();
txt_ApellidoMat.Clear();
txt_Fecha_Nac.Clear();
txt_Telefono.Clear();
```

Nos mostrara el mensaje de “El Cliente se ha Eliminado Correctamente”:

```
MessageBox.Show("El Cliente ha sido Eliminado Correctamente. ");
```

Añadimos un “DataGridView” para enlazar la base de datos con Windows Forms:



Ahora seleccionamos el botón de “Mostrar” en el formulario para que nos abra y agreguemos la codificación:

```
DataTable DT = new DataTable();
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;User ID=sa;Password=12345678;"))
{
    SqlDataAdapter Data = new SqlDataAdapter("Select * from Cliente", Conexion);
    Data.SelectCommand.CommandType = CommandType.Text; ;

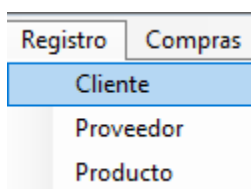
    Conexion.Open();
    Data.Fill(DT);

    dgv_Cliente.DataSource = DT;
}
```

Con esto ya estaría lista nuestro primer formulario ahora lo ejecutamos para ver si todo está correcto:



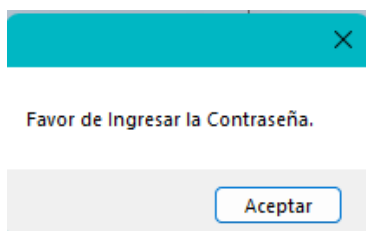
Seleccionamos en el menú la opción de “Registro” y “Cliente”:



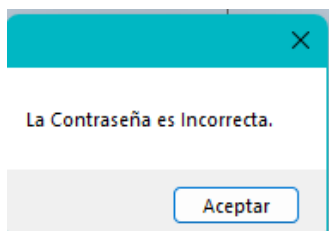
Como podemos ver nos arroja la ventana de “Ingreso” en el que tenemos que agregar una contraseña para lograr ingresar:



Si no ingresamos nada nos aparecerá el siguiente mensaje:



Si colocamos la incorrecta nos mandara el siguiente mensaje:



Si colocamos la contraseña correcta que es “963852741” nos abrirá la siguiente ventana que es en donde está alojada la base de “Registro de Clientes”:



Damos clic en “Mostrar” para que la base de datos se presente en la tabla que agregamos anteriormente:

	DNI	Nombre_Cliente	Apellido_Pat	Apellido_Mat	Fecha_Nacimiento	Telefono
▶	1	Sonia Alejandra	Fernandez	Moreno	2003-06-16	3366521589
	2	Daniela Femanda	Ordaz	Diaz	2001-02-25	3365214526
	3	Mario Alberto	Jimenez	Salcido	1995-07-01	3321458971
	4	Yesenia Guadalu...	Campos	Rojo	1998-12-12	3354892655
	5	Tamara Alejandra	Bernal	Ramos	2000-11-08	3322548965
	6	Samuel Enrique	Barrios	Enciso	2001-06-04	3365217789
	8	Angel Eduardo	Ortencio	Juarez	2002-05-25	3352658944
✱						

Colocaremos los datos para agregar a un cliente:

Datos Personales

Nombre(s):

Apellido Paterno:

Apellido Materno:

Fecha de Nacimiento (AAAA-MM-DD):

Telefono:

Damos clic en “Agregar” y nos mandara un aviso:

×

El Cliente ha sido Agregado Correctamente.

Aceptar

Presionamos el botón de “Mostrar” para actualizar la tabla:

Nombre_Cliente	Apellido_Pat	Apellido_Mat	Fecha_Nacimiento	Telefono
Mario Alberto	Jimenez	Salcido	1995-07-01	3321458971
Yesenia Guadalu...	Campos	Rojo	1998-12-12	3354892655
Tamara Alejandra	Bernal	Ramos	2000-11-08	3322548965
Samuel Enrique	Barrios	Enciso	2001-06-04	3365217789
Angel Eduardo	Ortencio	Juarez	2002-05-25	3352658944
Ricardo Fabian	Reyes	Caballero	1995-02-25	3325652145

Vamos a modificar la tabla, le cambiaremos la fecha de cumpleaños:

Datos Personales

Nombre(s):

Apellido Paterno:

Apellido Materno:

Fecha de Nacimiento (AAAA-MM-DD):

Telefono:

La fecha de cumpleaños antes era:

Ricardo Fabian	Reyes	Caballero	1995-02-25	3325652145
----------------	-------	-----------	------------	------------

Con el cambio antes que nada nos arroja un mensaje:

×

El Cliente ha sido Modificado Correctamente.

Seleccionamos “Mostrar” para ver el cambio:

Ricardo Fabian	Reyes	Caballero	2000-05-30	3325652145
----------------	-------	-----------	------------	------------

Si queremos eliminar al cliente solo colocamos su nombre y presionamos el botón de “Eliminar”:

Datos Personales

Nombre(s):

×

El Cliente ha sido Eliminado Correctamente.

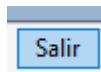
Actualizamos la tabla para revisar y en efecto este se eliminó por completo:

Nombre_Cliente	Apellido_Pat	Apellido_Mat	Fecha_Nacimiento	Telefono
Sonia Alejandra	Fernandez	Moreno	2003-06-16	3366521589
Daniela Fernanda	Ordaz	Diaz	2001-02-25	3365214526
Mario Alberto	Jimenez	Salcido	1995-07-01	3321458971
Yessenia Guadalu...	Campos	Rojo	1998-12-12	3354892655
Tamara Alejandra	Bernal	Ramos	2000-11-08	3322548965
Samuel Enrique	Barrios	Enciso	2001-06-04	3365217789
Angel Eduardo	Otencio	Juarez	2002-05-25	3352658944

Seleccionamos el botón de “Regresar” para que este nos mande al menú principal:



Por último, el botón de “Salir” que está en el menú para que nos cierre el programa por completo:



Vamos con el formulario de “Registro de Proveedor” damos doble clic y colocamos las librerías correspondientes:

```
using System.Data;
```

```
using System.Data.SqlClient;
```

Damos clic en el botón de agregar y colocamos la dirección de nuestro servidor:

```
1 referencia
private void btn_Agregar_Click(object sender, EventArgs e)
{
    using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;
```

Después de ello colocamos el código para que lo guarde la variable:

```
SqlCommand cmd = new SqlCommand("INSERT INTO Proveedor (Nombre, Dirección, Correo) VALUES ('"+ txt_Nombre_Proveedor.Text + "', '
cmd.CommandType = CommandType.Text;

veedor.Text + "', '"+ txt_Dirección_Proveedor.Text + "', '"+ txt_Correo.Text + "')", Conexion);
```

Colocamos el comando para que este lo ejecute:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

El código para limpiar todas las celdas cuando ya coloquemos la acción de “Agregar”:

```
txt_Nombre_Proveedor.Clear();
txt_Dirección_Proveedor.Clear();
txt_Correo.Clear();
```

El mensaje que nos mostrara cuando agreguemos un proveedor:

```
MessageBox.Show("El Proveedor se ha Agregado correctamente");
```

Ahora seleccionaremos el botón de “Modificar” y agregaremos la dirección:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza
```

Colocamos el código para que la variable lo guarde:

```
SqlCommand cmd = new SqlCommand("UPDATE Proveedor SET Dirección = '"+ txt_Dirección_Proveedor.Text + "', Correo
Correo = '"+ txt_Correo.Text + "', Where Nombre = '"+ txt_Nombre_Proveedor.Text + "' ", Conexion);
```

El Comando para que este lo ejecute:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

Para limpiar las “TextBox”:

```
txt_Nombre_Proveedor.Clear();
txt_Dirección_Proveedor.Clear();
txt_Correo.Clear();
```

El mensaje de “El Proveedor se ha Modificado Correctamente”:

```
MessageBox.Show("El Proveedor se ha Modificado Correctamente. ")
```

Vamos al botón de “Eliminar y colocamos la dirección de nuestro servidor:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza
```

El código que guardara la variable:

```
SqlCommand cmd = new SqlCommand("DELETE FROM Proveedor Where Nombre = '"+txt_Nombre_Proveedor.Text + "' ", Conexion);
```

El comando:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

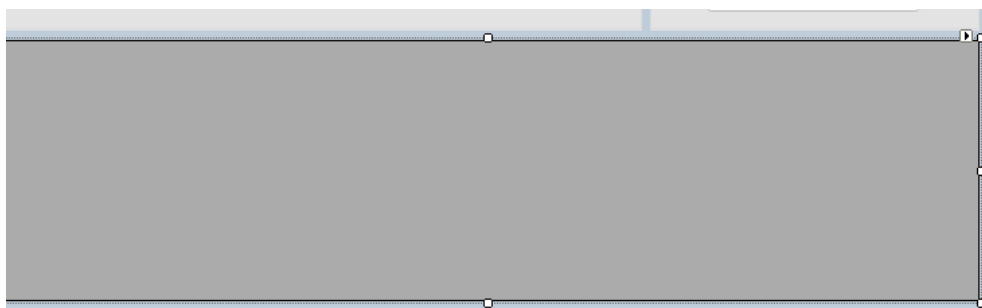

Para limpiar las “TextBox”:

```
txt_Nombre_Proveedor.Clear();
txt_Dirección_Proveedor.Clear();
txt_Correo.Clear();
```

El respectivo mensaje:

```
MessageBox.Show("El Proveedor se ha Eliminado Correctamente. ");
```

Agregamos el “DataGridView” en el formulario de “Registro de Proveedor”:



Ahora tenemos que seleccionar el botón de “Modificar” para agregar la codificación:

```
private void btn_Mostrar_Click(object sender, EventArgs e)
{
    DataTable DT = new DataTable();
    using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Espera
    {
        SqlDataAdapter Data = new SqlDataAdapter("Select * from Proveedor", Conexion);
        Data.SelectCommand.CommandType = CommandType.Text; ;

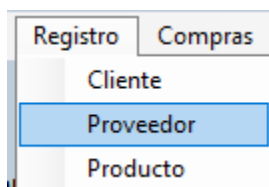
        Conexion.Open();
        Data.Fill(DT);

        dgv_Proveedor.DataSource = DT;
    }
}
```

Procederemos a ejecutar el programa para ver si está todo bien:



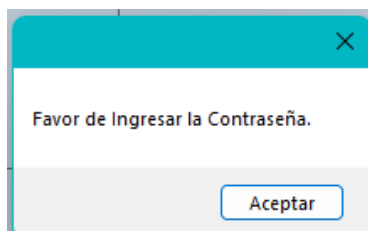
Seleccionamos el “Proveedor” que está en el menú:



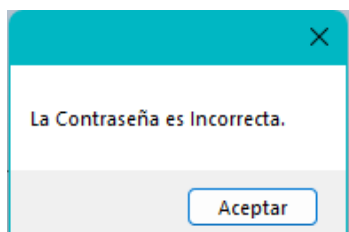
Al seleccionarlo nos abrirá el formulario de ingreso:



Si no colocamos nada nos mandara el siguiente mensaje:



Si colocamos mal la contraseña nos mandara el siguiente mensaje:



Si colocamos la contraseña “963852741” nos abrirá la sección de “Registro de Proveedor”:

Registro de Proveedor

Datos del Proveedor

Nombre:

Dirección:

Correo:

Opciones:

AGREGAR

MODIFICAR

MOSTRAR **ELIMINAR**

Regresar

Presionaremos el botón de “Mostrar” para ver si nos muestra la información de la tabla:

	NIF	Nombre	Dirección	Correo
▶	1	NIKE	LERDO DE TEJ...	nike@gmail.com
	2	PUMA	FCO I MADERO ...	puma@gmail.com
	3	PIRMA	REVOLUCION 2....	pima@gmail.com
	4	CONVERSE	REFORMA 304 ...	converse@gmail....
	5	ADIDAS	PORFIRIO DIAZ ...	adidas@gmail.com
*				

Ahora agregaremos a un nuevo proveedor:

Datos del Proveedor

Nombre:

Dirección:

Correo:

Seleccionamos el botón de “Agregar”:

×

El Proveedor se ha Agregado correctamente

Aceptar

Seleccionamos el botón de “Mostrar” y podemos apreciar que se agregó correctamente:

	NIF	Nombre	Dirección	Correo
▶	1	NIKE	LERDO DE TEJ...	nike@gmail.com
	2	PUMA	FCO I MADERO ...	puma@gmail.com
	3	PIRMA	REVOLUCION 2,...	pima@gmail.com
	4	CONVERSE	REFORMA 304 ...	converse@gmail....
	5	ADIDAS	PORFIRIO DIAZ ...	adidas@gmail.com
	6	NIKE	LA TEJADA 253,...	NIKE1@GMAIL....
*				

Ahora modificaremos el proveedor, cambiaremos su correo:

Datos del Proveedor

Nombre:

Dirección:

Correo:

Nos muestra el mensaje de la codificación:



El Proveedor se ha Modificado Correctamente.

Observamos la modificación:

6	NIKE	LA TEJADA 253,...	nike21@gmail.com
---	------	-------------------	------------------

Para eliminar solo colocamos el nombre del Proveedor y sucederá algo curioso, se van a eliminar todos los proveedores de la marca “NIKE” por qué solo agregaremos el nombre:

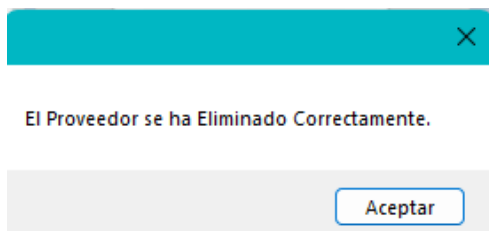
Datos del Proveedor

Nombre:

Dirección:

Correo:

Nos mandara el siguiente mensaje:



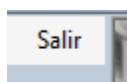
Seleccionamos “Mostrar” y veamos lo que ocurre, como podemos observar todos los registros de “NIKE” se eliminaron por completo:

NIF	Nombre	Dirección	Correo
2	PUMA	FCO I MADERO ...	puma@gmail.com
3	PIRMA	REVOLUCION 2,...	pima@gmail.com
4	CONVERSE	REFORMA 304 ...	converse@gmail....
5	ADIDAS	PORFIRIO DIAZ ...	adidas@gmail.com

Presionamos el botón de “Regresar” para que nos mande al menú principal:



Y por último salir:



Vamos al formulario de “Registro de Productos” y agregamos las librerías correspondientes:

```
using System.Data;
```

```
using System.Data.SqlClient;
```

Después de ello daremos clic al botón de “Agregar” para colocar la dirección de nuestro servidor:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;In
```

El código:

```
SqlCommand cmd = new SqlCommand("INSERT INTO Producto ( Género, Marca, Tipo, Talla) VALUES ('"+txt_Genero.Text + "',  
cmd.CommandType = CommandType.Text;  
('"+txt_Genero.Text + "', '"+txt_Marca.Text + "', '"+txt_Tipo.Text + "', '"+txt_Talla.Text + "')", Conexion);
```

El comando para ejecutarlo:

```
cmd.CommandType = CommandType.Text;  
Conexion.Open();  
cmd.ExecuteNonQuery();
```

Para limpiar las “TextBox”:

```
txt_Genero.Clear();  
txt_Marca.Clear();  
txt_Tipo.Clear();  
txt_Talla.Clear();
```

Después el mensaje correspondiente:

```
MessageBox.Show("El Producto se ha Agregado correctamente");
```

Seguimos con el botón de “Modificar”, lo seleccionamos y colocamos la dirección de nuestro servidor:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;In
```

El código de nuestra acción:

```
SqlCommand cmd = new SqlCommand("UPDATE Producto SET Marca = '" + txt_Marca.Text + "', Tipo = '" + txt_Tipo.Text + "', T  
cmd.CommandType = CommandType.Text;  
'" + txt_Tipo.Text + "', Talla = '" + txt_Talla.Text + "' Where Género = '" + txt_Genero.Text + "' ", Conexion);
```

El comando para ejecutarlo:

```
cmd.CommandType = CommandType.Text;  
Conexion.Open();  
cmd.ExecuteNonQuery();
```

Para limpiar las “TextBox”:

```
txt_Genero.Clear();
txt_Marca.Clear();
txt_Tipo.Clear();
txt_Talla.Clear();
```

Ahora que nos muestre el mensaje correspondiente:

```
MessageBox.Show("El Producto se ha Modificado Correctamente. ");
```

Nos dirigimos al botón de eliminar y colocamos la dirección de nuestro servidor:

```
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza;I
{
```

El código:

```
SqlCommand cmd = new SqlCommand("DELETE FROM Producto Where Tipo = '" + txt_Tipo.Text + "'", Conexion);
```

El comando para que se ejecute:

```
cmd.CommandType = CommandType.Text;
Conexion.Open();
cmd.ExecuteNonQuery();
```

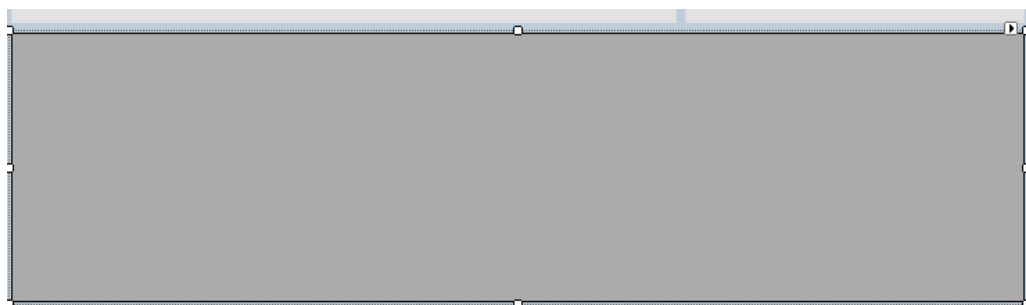
El código para limpiar las “TextBox”:

```
txt_Genero.Clear();
txt_Marca.Clear();
txt_Tipo.Clear();
txt_Talla.Clear();
```

Con el mensaje correspondiente:

```
MessageBox.Show("El Producto se ha Eliminado Correctamente. ");
```

Agregamos una “DataGridView” al formulario de “Registro de Productos”:



Seleccionamos el botón de “Mostrar” para agregar la codificación:

```
DataTable DT = new DataTable();
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la_Esperanza
{
    SqlDataAdapter Data = new SqlDataAdapter("Select * from Productos", Conexion);
    Data.SelectCommand.CommandType = CommandType.Text; ;

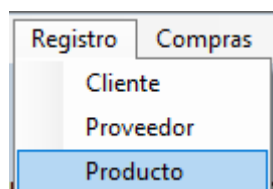
    Conexion.Open();
    Data.Fill(DT);

    dgv_Productos.DataSource = DT;
}
```

Con esto será suficiente, ahora ejecutaremos el programa para validar:



Seleccionamos “Producto” que está en el menú:



Después de ello nos arrojará la ventana de “Ingreso” de nuestro formulario:



Si no colocamos nada nos aparece esto:

×

Favor de Ingresar la Contraseña.

Aceptar

Si colocamos la contraseña incorrecta nos arroja lo siguiente:

×

La Contraseña es Incorrecta.

Aceptar

Si colocamos la contraseña “963852741” nos mandara al siguiente formulario:

Registro de Productos

Datos de los Productos

Género:

Marca:

Tipo:

Talla:

Opciones

AGREGAR

MODIFICAR

MOSTRAR
ELIMINAR

Regresar

Daremos clic al botón de “Mostrar” para ver la tabla:

	Codigo	Género	Marca	Tipo	Talla
▶	1	Mujer	NIKE	Botas altas	23
	2	Mujer	NIKE	Botas australianas	23
	3	Mujer	PUMA	Botines	23.5
	4	Mujer	PUMA	Merceditas	24
	5	Mujer	CONVERSE	Zapatos de salón	24.5
	6	Mujer	CONVERSE	Sandalias de Tac...	25
	7	Mujer	PIRMA	Cuñas	25.5
	8	Mujer	PIRMA	Mary Jane	26

Agregaremos un nuevo producto de dama:

Datos de los Productos

Género:

Marca:

Tipo:

Talla:

Damos clic en “Agregar”:



El Producto se ha Agregado correctamente

Actualizamos la tabla y veamos el cambio:

20	Hombre	ADIDAS	Sandalias	30
21	Mujer	NIKE	Deportivo	23.5

Modificaremos la talla del producto:

Datos de los Productos

Género:

Marca:

Tipo:

Talla:

Damos clic en “Modificar”:



El Producto se ha Modificado Correctamente.

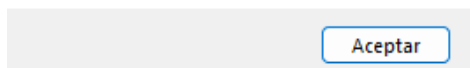
Ahora actualizamos la tabla para revisar:

21	Mujer	NIKE	Deportivo	24
----	-------	------	-----------	----

Ahora para eliminar colocaremos el tipo de calzado y presionamos el botón de “Eliminar”:



El Producto se ha Eliminado Correctamente.



20	Hombre	ADIDAS	Sandalias	30
----	--------	--------	-----------	----

Presionamos el botón de “Regresar” y este nos mandara al menú principal:

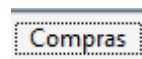


Para seleccionar el botón de salir y cerrar el programa:



Ahora creamos un nuevo formulario para agregarlo a “Compra”, empezaremos con el diseño:

Crearemos una clase para darle acciones a nuestros comandos. Nos dirigimos al menú y seleccionamos “Compras” para vincularla al formulario actual:



Nos abrirá una ventana en la que estaremos colocando el código de cada acción:

```
0 referencias
public class Clase_Compra
{
    0 referencias
    public float Sumar ( float numero1, float numero2)
    {
        float resultado;
        resultado = numero1 + numero2;
        return resultado;
    }

    0 referencias
    public float Restar ( float numero1, float numero2)
    {
        float resultado;
        resultado = numero1 - numero2;
        return resultado;
    }

    0 referencias
    public float Multiplicar ( float numero1, float numero2)
    {
        float resultado;
        resultado = numero1 * numero2;
        return resultado;
    }

    0 referencias
    public float Dividir ( float numero1, float numero2)
    {
        float resultado;
        resultado = numero1 / numero2;
        return resultado;
    }
}
```

Ahora seleccionaremos el botón de “Sumar” para agregar la codificación correspondiente:

```
private void btn_Sumar_Click(object sender, EventArgs e)
{
    float resultado;
    Clases.Clase_Compra clase_Compra = new Clases.Clase_Compra();
    resultado = clase_Compra.Sumar(float.Parse(txt_Numero1.Text), float.Parse(txt_Numero2.Text));
    txt_Resultado.Text = resultado.ToString();
}
```

Hacemos lo mismo con el botón de “Resta”:

```
float resultado;
Clases.Clase_Compra clase_Compra = new Clases.Clase_Compra();
resultado = clase_Compra.Restar(float.Parse(txt_Numero1.Text), float.Parse(txt_Numero2.Text));
txt_Resultado.Text = resultado.ToString();
```

Botón de “Multiplicar”:

```
float resultado;
Clases.Clase_Compra clase_Compra = new Clases.Clase_Compra();
resultado = clase_Compra.Multiplicar(float.Parse(txt_Numero1.Text), float.Parse(txt_Numero2.Text));
txt_Resultado.Text = resultado.ToString();
```

Por ultimo con el de “División”:

```
float resultado;
Clases.Clase_Compra clase_Compra = new Clases.Clase_Compra();
resultado = clase_Compra.Dividir(float.Parse(txt_Numero1.Text), float.Parse(txt_Numero2.Text));
txt_Resultado.Text = resultado.ToString();
```

Escondemos la “TextBox” de resultado para que no puedan colocar datos en ese lugar:

```
InitializeComponent();
txt_Resultado.Enabled = false;
```

Vamos a darle acción al botón de “Regresar”:

```
this.Close();
Form_Menú back = new Form_Menú();
back.Show();
```

Vamos a limpiar la sección de “TextBox”:

```
1 referencia
private void btn_Limpiar_Click(object sender, EventArgs e)
{
    txt_Numero1.Clear();
    txt_Numero2.Clear();
    txt_Resultado.Clear();
}
```

Damos la codificación a la tabla de base de datos para que nos muestre la información, recordar que tenemos que agregar las librerías:

```
using System.Data;
```

```
using System.Windows.Forms;
using System.Data.SqlClient;
```

Empezamos con la codificación:

```
DataTable DT = new DataTable();
using (SqlConnection Conexion = new SqlConnection("Data Source=ROG-ONE\\SQLEXPRESS;Initial Catalog=Zapateria_la
{
    SqlDataAdapter Data = new SqlDataAdapter("Select * from Compra", Conexion);
    Data.SelectCommand.CommandType = CommandType.Text; ;

    Conexion.Open();
    Data.Fill(DT);

    dgv_Compra.DataSource = DT;
}
```

Por ultimo vamos a enlazar los formularios, seleccionamos la sección de “Compra”:

Compras

Nos abrirá la codificación:

```
Formularios.Form_Compra form_Compra = new Formularios.Form_Compra();
form_Compra.Show();
this.Hide();|
```

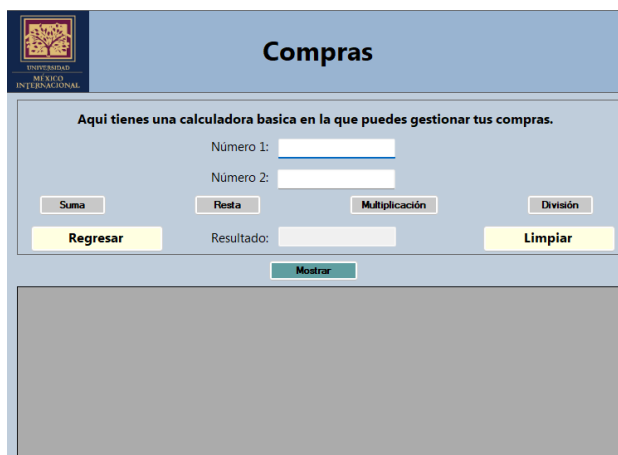
Ahora ejecutaremos el programa para ver si todo funciona correctamente:



Seleccionamos la sección de “Compras” para visualizar el formulario:

Compras

Nos abre el siguiente formulario (IMPORTANTE MENCIONAR QUE PARA ESCRIBIR LOS DECIMALES SEA CON COMAS Y NO PUNTOS):



Compras

Aquí tienes una calculadora básica en la que puedes gestionar tus compras.

Número 1:

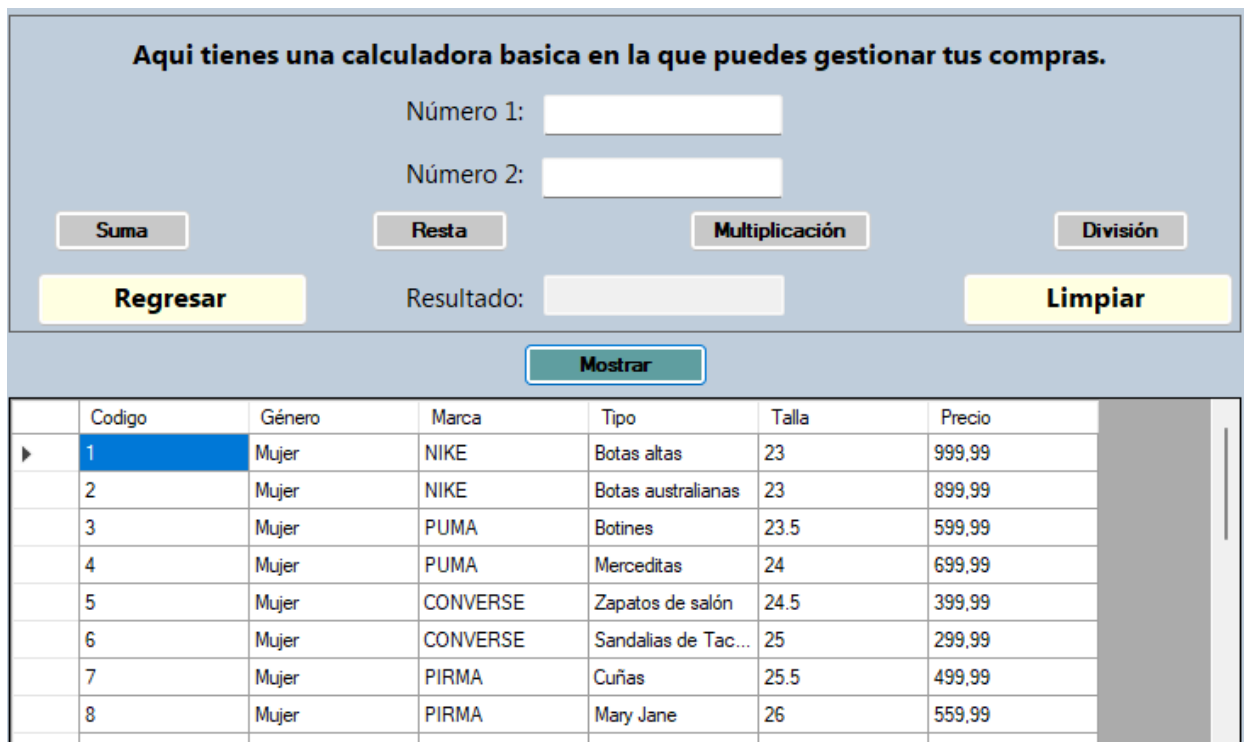
Número 2:

Suma Resta Multiplicación División

Regresar Resultado: Limpiar

Mostrar

Seleccionaremos el botón de “Mostrar” para que nos muestre la base de datos y podamos crear cálculos con el:



Aquí tienes una calculadora básica en la que puedes gestionar tus compras.

Número 1:

Número 2:

Suma Resta Multiplicación División

Regresar Resultado: Limpiar

Mostrar

	Codigo	Género	Marca	Tipo	Talla	Precio
▶	1	Mujer	NIKE	Botas altas	23	999,99
	2	Mujer	NIKE	Botas australianas	23	899,99
	3	Mujer	PUMA	Botines	23.5	599,99
	4	Mujer	PUMA	Merceditas	24	699,99
	5	Mujer	CONVERSE	Zapatos de salón	24.5	399,99
	6	Mujer	CONVERSE	Sandalias de Tac...	25	299,99
	7	Mujer	PIRMA	Cuñas	25.5	499,99
	8	Mujer	PIRMA	Mary Jane	26	559,99

Sumaremos dos productos para ver el precio:

1	Mujer	NIKE	Botas altas	23	999,99
2	Mujer	NIKE	Botas australianas	23	899,99

Colocamos los datos y damos clic en el botón de “Sumar”:

Número 1:

Número 2:

Resultado:

Damos clic en el Botón de Limpiar para que este borre los datos:

Número 1:

Número 2:

Resultado:

Con esto estás listo para gestionar las compras, ahora solo damos clic en “Regresar para que nos mande al menú principal:



Por último, el botón de “Salir” para finalizar con este proyecto final:

Conclusión

Que puedo decir de esta actividad final, fue una de las actividades más largas y abrumadoras que me ha tocado hacer, el saber cómo acomodar todo y crear el diseño era lo que poco a poco me vencían, pero no cedi, e inclusive hubo un punto en el que había avanzado tanto, empecé desde la mañana y eran las 5 de la tarde, tenía tantos programas abiertos en la laptop que sin ningún aviso se congeló, no reacciono y tuve que apagarla, solo para darme cuenta de que no había guardado nada en casi tres horas de trabajo, cuando encendí la laptop revise los documentos de “Word”, de “Visual Studio” y en efecto había perdido casi tres horas de trabajo todo por no guardar constantemente los archivos, pero vamos, valió la pena, aquí está, este es el documento que tiene la actividad pedida por el profesor, puede haber partes en las que el documento tenga explicaciones muy cortas, eso es debido a que en ese punto tuve que trabajar lo más rápido para recuperar el progreso perdido.

Link de GitHub

<https://github.com/Leyzu-Ing/Lenguaje-de-Programaci-n-III.git>

Referencias

rwestMSFT. (s. f.). *¿Qué es SQL Server? - SQL Server*. Microsoft Learn.

<https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver16>