



CanoMontoya_PR1x4

Rubén Cano Montoya

Contenido

Diseño.....	4
Ventana 1 gestor de operaciones CRUD.....	4
Ventana 2 Lista de Registros	4
BBDD	5
Campos	5
Funcionalidad Agregar	5
Errores.....	5
Agregar sin datos en los campos.....	6
Agregar registro ya creado.....	6
Agregar Registro con tipo de datos incorrectos	6
Agregar registro con precio decimal con coma.....	7
Código.....	7
Agregar decimal con coma	9
Agregar Registro Ya Creado.....	9
Funcionalidad Eliminar.....	9
Errores.....	10
Eliminación de producto sin Id.	10
Eliminación de registro inexistente.....	11
Eliminación de Producto de Id como cadena de texto	11
Código.....	11
Funcionalidad Modificar.....	13
Errores.....	15
Modificación de registro con campo vacío	15
Modificación de registro inexistente.....	15
Modificación de registro de Id de tipo incorrecto	16
Código.....	16
Funcionalidad Consultar	18
Errores.....	19
Consulta de registro de Id de tipo incorrecto	19
Consulta de registro sin Id	19
Consulta de registro Inexistente.....	20
Código.....	20
Funcionalidad Imagen.....	22
Subir Imagen.....	22
Código.....	23
Quitar Imagen	23
Código.....	23

Funcionalidad Limpiar Datos	23
Código.....	24
Funcionalidad listar registros.....	24
Código.....	24
Filtro	25
Error.....	25
Enviar registro a gestor CRUD	26
Codigo.....	27
Funcionalidad Ir Atras	27
Código.....	28

Diseño

Ventana 1 gestor de operaciones CRUD

El Corte Finlandes — ✕

Gestor Base De Datos

Id Producto: Descripción:

Departamento: Stock:

Imagen:  Precio:

Acciones:

- Agregar (+)
- Eliminar (🗑️)
- Consultar (🔍)
- Modificar (✎️)
- Salir

Ventana 2 Lista de Registros

El Corte Finlandes — ✕

Filtrar por: Valor:

	Id	Descripcion	Precio	Departamento	NumeroDeExistencias	Imagen
▶	2	Nintendo DS	55,35	Electronica	2	
	3	GameBoy	10	Electronica	3	
	4	al día	23	Alimentacion	3	

BBDD

Campos

T4\PC03\SQLEXPRESS...s - dbo.Inventario			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	Descripcion	nvarchar(50)	<input type="checkbox"/>
	Precio	float	<input type="checkbox"/>
	Departamento	nvarchar(12)	<input type="checkbox"/>
	Imagen	image	<input checked="" type="checkbox"/>
	NumeroDeExistencias	int	<input type="checkbox"/>

La base de datos tiene 6 campos incluyendo la clave primaria Id, el precio en float, la imagen la cual se va a insertar en una secuencia de bytes y finalmente el departamento el cual en el programa se va a permitir tres tipos de departamentos para su inserción, como se puede ver la imagen es la única que va a aceptar nulos, es decir, va a ser un campo opcional.

Departamento

Ropa

Alimentacion

Electronica

Funcionalidad Agregar

El Corte Finlandes

Gestor Base De Datos

Id Producto

1

Descripción

Jersey

Departamento

Ropa

Stock

20

Imagen



Precio

15,3

Subir Imagen

Quitar Imagen

Limpiar Datos

Ver Registros

Agregar

Eliminar

Consultar

Modificar

Salir

Es obligatoria la introducción de todos los campos menos la imagen, para la inserción de un producto.

	Id	Descripcion	Precio	Departamento	Imagen	NumeroDeExistencias
1	1	Bufanda	9,5	Ropa	0x89504E47D0A1A0A0000000D49484452000001E0000002...	24
2	2	Nintendo DS	55,35	Electronica	NULL	2

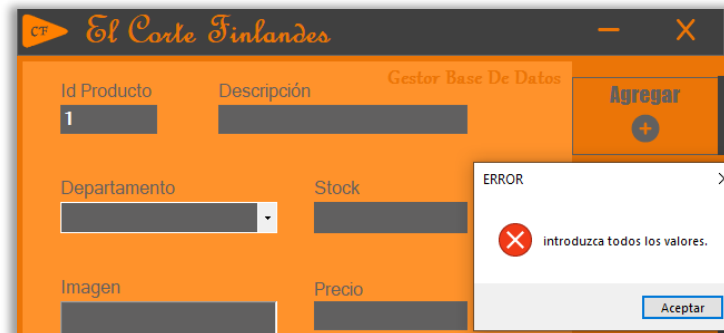
Como se puede ver una vez hemos pulsado el botón de agregar se ha insertado el correspondiente registro en la base de datos.

Errores

Se han tratado de asegurar los siguientes errores.

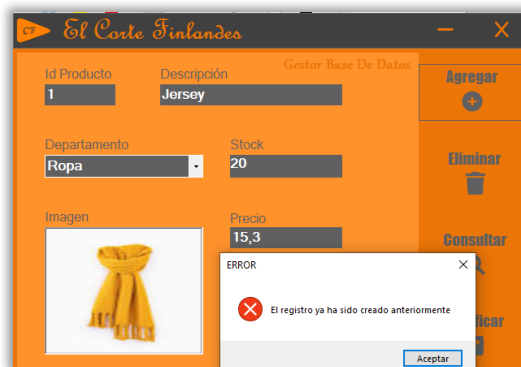
Agregar sin datos en los campos

El primero sería al intentar agregar un registro sin valores en los campos, como se ha indicado anteriormente solo se permitirá la inserción de los registros con contenido en todos los campos.



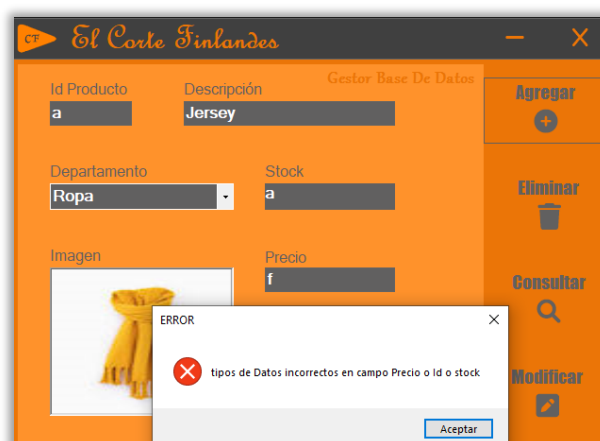
Agregar registro ya creado

Se va a controlar que cuando se intente agregar un producto de forma repetida se indique, se evite y lo indique.



Agregar Registro con tipo de datos incorrectos

Se va a controlar que a la hora de introducir los datos además de que se hayan rellenado todos que sean los tipos de datos incorrectos para cada campo.



Agregar registro con precio decimal con coma

Como tal la base de datos acepta puntos a la hora de indicar el valor de un decimal para hacer un Insert, es por eso que se ha tratado de aceptar comas a la hora de introducir un registro en la aplicación.



Código

```

//////////AGREGAR//////////
private void iconButton1_Click(object sender, EventArgs e)
{
    //se crea un capturador de errores para que en cualquier error no cierre la aplicacion.
    try
    {
        //se abre la conexion
        conexion.Open();
        //evalua que todos los campos tengan contenido y si no lo tiene lo indica.
        if (obtenerDepartamento() != null && !textBox1.Text.Equals("") && !textBox2.Text.Equals("") && !textBox3.Text.Equals("") && !textBox4.Text.Equals(""))
        {
            if (obtenerCampos())
            {
                //se cogen los valores, el contenido introducido en dichos campos.

                //Se guarda en una cadena la query que va a insertar los datos en la tabla.
                string cadena = "insert into Inventario (Id, Precio, Descripcion, Departamento, NumeroDeExistencias) values"+
                "(" + Id + ", " + Precio + ", " + Descripcion + ", " + departamento + ", " + stock + " )";
                //creo un comando con la cadena que hemos guardado antes y la conexion a la base de datos.
                SqlCommand comando = new SqlCommand(cadena, conexion);

                /*comprueba si ya existe un registro con el id proporcionado,
                en el caso de que no lo haya ejecuta el comando anteriormente creado
                y limpia todos los campos menos el de imagen puesto que todavia no ha terminado la insercion.
                en el caso de que ya halla un registro lo indica mediante un MessageBox
                */
                if (!comprobacionRegistros())
                {
                    MessageBox.Show("Registro agregado con exito", "Registro Agregado", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    textBox1.Text = "";
                    textBox2.Text = "";
                    textBox3.Text = "";
                    textBox4.Text = "";
                    comboBox1.SelectedIndex = -1;

                    comando.ExecuteNonQuery();//inserta los valores.
                    if (pictureBox2.Image != null)
                    {
                        /*
                        * una vez insertados los datos se va a evaluar si se va a introducir la imagen o no,
                        * evaluando el contenido del pictureBox
                        */
                        SqlCommand ComandoMeterImg = new SqlCommand();
                        ComandoMeterImg.Connection = conexion;
                        /*una vez creado el comando y agregada su conexion
                        * introduciremos la query que va a insertar la imagen
                        * en el registro anteriormente creado mediante parametros.
                        */
                        ComandoMeterImg.CommandText = "Update Inventario Set Imagen=@Foto where Id=" + Id;
                        ComandoMeterImg.Parameters.Add("@Foto", System.Data.SqlDbType.Image);//se agrega el comando y se indica de que tipo va a ser.
                        /*creamos el objeto de flujo que va recoger el dato de la imagen mediante el metodo .Save .
                        * posteriormente le indicamos al comando que el parametro tiene como valor el buffer del flujo.
                        * ejecutamos la query y vaciamos el pictureBox.
                        */
                        MemoryStream stream = new MemoryStream();

                        pictureBox2.Image.Save(stream, System.Drawing.Imaging.ImageFormat.Png);

                        ComandoMeterImg.Parameters["@Foto"].Value = stream.GetBuffer();
                        ComandoMeterImg.ExecuteNonQuery();
                        pictureBox2.Image = null;
                    }
                }
            }
            else
            {
                MessageBox.Show("El registro ya ha sido creado anteriormente", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("tipos de Datos incorrectos en campo Precio o Id o stock", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("introduzca todos los valores.", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    conexion.Close();
}
catch (Exception ex)
{
    conexion.Close();
    MessageBox.Show(ex.ToString());
}
}

```


Agregar decimal con coma

Se ha usado la función .Replace() dentro del método obtenerCampos esta función mete el valor de los campos dentro de las variables globales, y devuelve el estado booleano del intento de pasar los valores recibidos en sus respectivos tipos.

```
4 Referencias
public bool obtenerCampos()
{
    departamento = obtenerDepartamento();
    Id = textBox1.Text;
    Description = "" + textBox2.Text + "";
    Precio = textBox3.Text.Replace('.', ',');
    stock = textBox4.Text;
    int a;
    double b;
    return (int.TryParse(Id, out a) && double.TryParse(Precio, out b) && int.TryParse(stock, out a));
}
```

Agregar Registro Ya Creado

Para evaluar si ya hay un registro antes de crearlo se ejecuta el método ComprobacionRegistros(), que hace una consulta ha dicho registro si es true, es que ya existe por lo tanto debe saltar el error de que ya de ha creado

```
1 Referencias
public Boolean comprobacionRegistros()
{
    SqlConnection conexion1 = new SqlConnection("Server = localhost\\SQLEXPRESS; Database = ElCorteFinlandes; integrated security = true");
    conexion1.Open();
    SqlCommand comandoRev = new SqlCommand("select Id from inventario where Id=" + textBox1.Text, conexion1);
    SqlDataReader sDT = comandoRev.ExecuteReader();
    Boolean hayRegistros = sDT.Read();

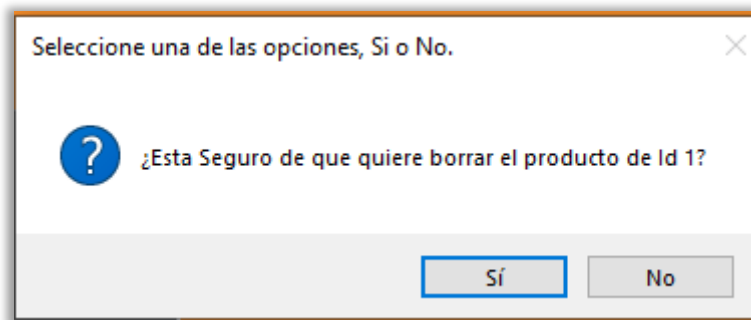
    return hayRegistros;
}
```

Funcionalidad Eliminar

The screenshot shows a Windows application titled "El Corte Finlandes" with a "Gestor Base De Datos" (Database Manager) window. The window has a light orange background and contains the following elements:

- Product Information:**
 - Id Producto:** 1
 - Descripción:** Bufanda
 - Departamento:** Ropa (selected from a dropdown)
 - Stock:** 24
 - Precio:** 9,5
 - Imagen:** A yellow scarf image is displayed in a white box.
- Actions:**
 - Subir Imagen:** Button to upload an image.
 - Quitar Imagen:** Button to remove the image.
 - Limpiar Datos:** Button to clear the data.
 - Ver Registros:** Button to view records, accompanied by a database cylinder icon.
- Navigation/Management:**
 - Agregar:** Button with a plus icon to add a new record.
 - Eliminar:** Button with a trash can icon to delete a record.
 - Consultar:** Button with a magnifying glass icon to search for records.
 - Modificar:** Button with a pencil icon to edit a record.
 - Salir:** Button to exit the application.

Para la eliminación de un producto solo es necesario la introducción del campo Id.



Tras pulsar en Sí en el aviso, hemos borrado el registro.

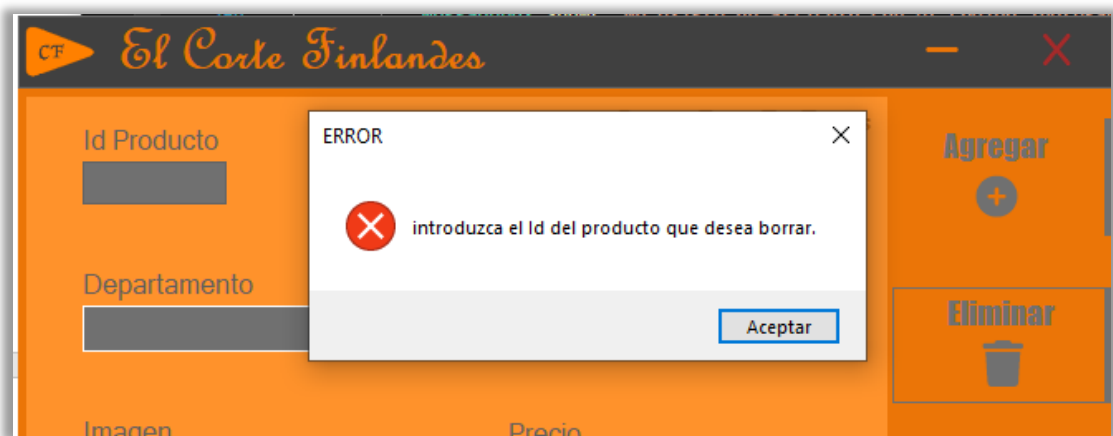
A screenshot of a table with columns: Id, Descripción, Precio, Departamento, Imagen, and NumeroDeExistencias. The table has three rows. The first row has Id 2, Descripción Nintendo DS, Precio 55,35, Departamento Electronica, Imagen NULL, and NumeroDeExistencias 2. The second row has Id 3, Descripción gamevoby, Precio 4,56, Departamento Electronica, Imagen NULL, and NumeroDeExistencias 3. The third row has Id 4, Descripción afdra, Precio 23, Departamento Alimentacion, Imagen 0x89504E470D0A1A0A000000D49484452000004EC000002..., and NumeroDeExistencias 3. The first row is highlighted in yellow.

	Id	Descripción	Precio	Departamento	Imagen	NumeroDeExistencias
1	2	Nintendo DS	55,35	Electronica	NULL	2
2	3	gamevoby	4,56	Electronica	NULL	3
3	4	afdfa	23	Alimentacion	0x89504E470D0A1A0A000000D49484452000004EC000002...	3

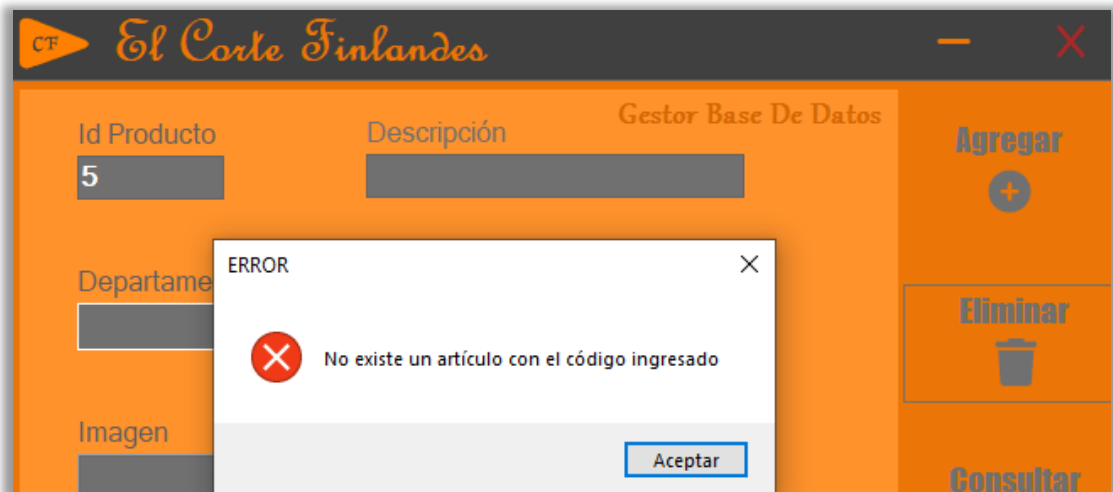
Como se puede ver una vez se ha clicado en el botón de eliminar ya no aparece el registro, pues se ha eliminado.

Errores

Eliminación de producto sin Id.

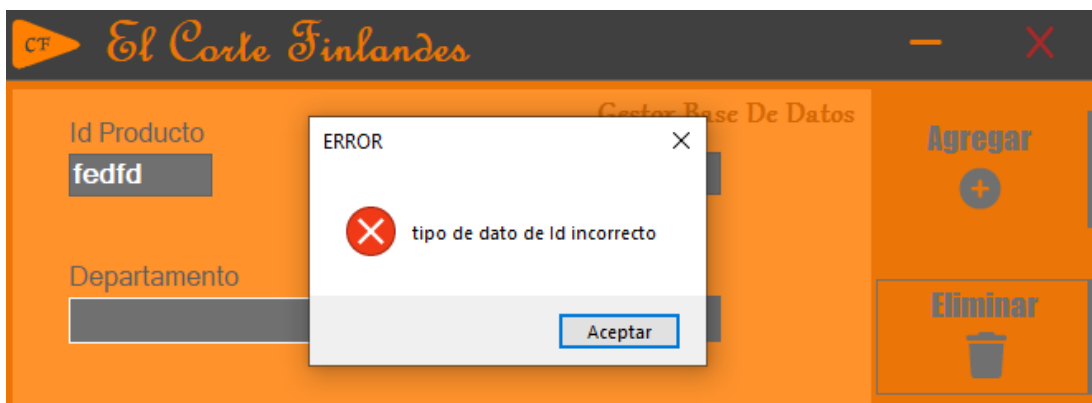


Eliminación de registro inexistente



Eliminación de Producto de Id como cadena de texto

Al igual que en el caso anterior se ha controlado el caso en el que el usuario introduzca como Id del producto a borrar una cadena de texto.



Código

```

//////////ELIMINAR//////////
private void iconButton4_Click(object sender, EventArgs e)
{
    try
    {
        conexion.Open();//se abre la conexion
        //se evalua si se ha ingresado algun valor en el campo del Id, si no se ha hecho lo indica.
        if (!textBox1.Text.Equals(""))
        {
            //se obtienen los valores de los campos.
            obtenerCampos();
            int x;
            if (int.TryParse(Id, out x))
            {
                /*se indica mediante una advertencia si esta seguro de que quiere borrar el producto
                * si elige que no, se indica que no se ha borrado, y si elige que si se borrará.
                */

                DialogResult mensaje = MessageBox.Show("¿Esta Seguro de que quiere borrar el producto de Id " + Id + "?",
                "Seleccione una de las opciones, Si o No.", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (mensaje == DialogResult.No)
                {
                    MessageBox.Show("La operacion ha sido cancelada", "Operacion Abortada", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                else
                {
                    string cadena = "delete from Inventario where Id=" + Id;
                    SqlCommand comando = new SqlCommand(cadena, conexion);//Crea el comando que va a ejecutar la query cadena sobre la conexion(BBDD).
                    if (comando.ExecuteNonQuery() == 1)
                    {
                        //si la ejecucion ha afectado a al menos una fila se indicará que el producto ha sido borrado y si no se indicará que no existe.
                        textBox1.Text = "";
                        textBox2.Text = "";
                        textBox3.Text = "";
                        textBox4.Text = "";
                        comboBox1.SelectedIndex = -1;
                        pictureBox2.Image = null;
                        MessageBox.Show("Articulo Borrado", "El articulo ha sido borrado", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                    else
                    {
                        MessageBox.Show(" No existe un artículo con el código ingresado ", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
        }
        else
        {
            MessageBox.Show("tipo de dato de Id incorrecto", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("introduzca el Id del producto que desea borrar.", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    conexion.Close();
}
catch (Exception ex)
{
    conexion.Close();
    MessageBox.Show(ex.ToString());
}
}

```

Funcionalidad Modificar

El Corte Finlandes Gestor Base De Datos

Id Producto: 3 Descripción: GameBoy

Departamento: Electronica Stock: 3

Imagen: [Placeholder] Precio: 4,56

Subir Imagen

Quitar Imagen

Limpiar Datos

Ver Registros

Agregar, Eliminar, Consultar, Modificar, Salir

	Id	Descripcion	Precio	Departamento	Imagen	NumeroDeExistencias
1	2	Nintendo DS	55,35	Electronica	NULL	2
2	3	GameBoy	4,56	Electronica	NULL	2

En este caso se va a modificar el campo de precio del producto GameBoy de casi 5 euros, a 10 euros.

El Corte Finlandes Gestor Base De Datos

Id Producto: 3 Descripción: GameBoy

Departamento: Electronica Stock: 3

Imagen: [Placeholder] Precio: 10

Subir Imagen

Quitar Imagen

Limpiar Datos

Ver Registros

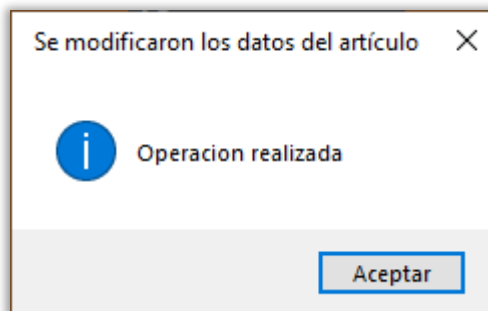
Agregar, Eliminar, Consultar, Modificar, Salir

Selecione una de las opciones, Si o No.

¿Esta Seguro de que quiere modificar el producto de Id 3?

Sí No

Se solicita previamente la confirmación de la modificación, y en el caso de que se pulse en "Si" se modificará el registro en cuestión y se indicará con su correspondiente aviso de información.



	Id	Descripcion	Precio	Departamento	Imagen	NumeroDeExistencias
1	2	Nintendo DS	55.35	Electronica	NULL	2
2	3	GameBoy	10	Electronica	NULL	3
3	4	afdfa	23	Alimentacion	0x89504E470D0A1A0A0000000D49484452000004EC000002...	3

Dado que la imagen es un campo opcional se ha tenido en cuenta si se ha insertado una imagen o se ha quitado para que a la hora de modificar el registro se actualice el campo de imagen.

De forma que si modifico un producto y le subo una imagen a la hora de clicar en modificar se evaluara si hay una imagen subida y la agrega al registro mediante Update.

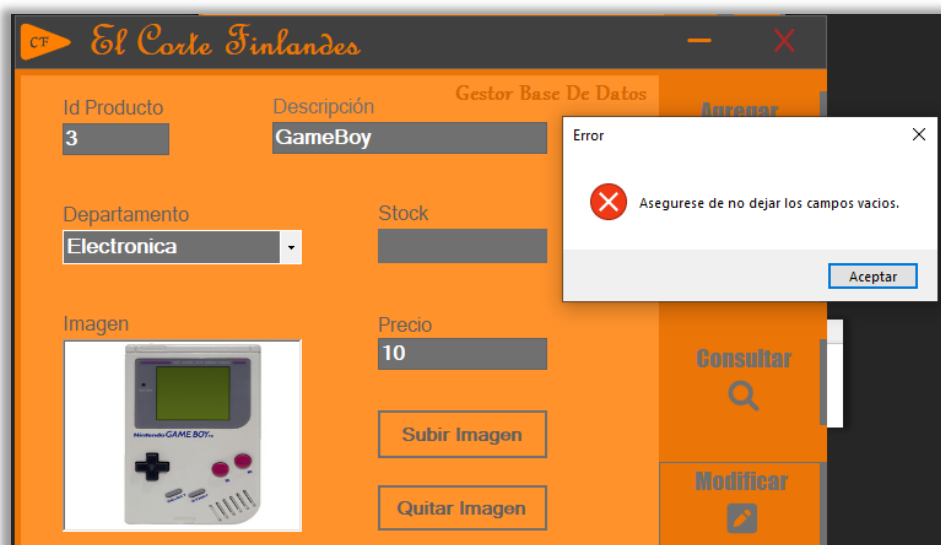


Results		Messages				
	Id	Descripcion	Precio	Departamento	Imagen	NumeroDeExistencias
1	2	Nintendo DS	55.35	Electronica	NULL	2
2	3	GameBoy	10	Electronica	0x89504E470D0A1A0A0000000D494844520000012C000001...	3
3	4	afdfa	23	Alimentacion	0x89504E470D0A1A0A0000000D49484452000004EC000002...	3

Errores

Modificación de registro con campo vacío

Dado que solo se permite nulos para el campo imagen para los demás campos se evalúa que tengan contenido, en el caso de que falte contenido en cualquier campo, se avisará mediante un messageBox.



Modificación de registro inexistente



En el caso de que se indique un Id inexistente se indicara mediante un messageBox

Modificación de registro de Id de tipo incorrecto

The screenshot shows a web application titled "El Corte Finlandes" with a "Gestor Base De Datos" (Database Manager) interface. The form contains the following fields:

- Id Producto:** sdf
- Descripción:** afdfa
- Departamento:** Alimentacion (dropdown menu)
- Stock:** 3
- Imagen:** A placeholder image showing silhouettes of people.
- Precio:** 23

On the right side of the form, there are four buttons: "Agregar" (Add), "Eliminar" (Delete), "Consultar" (Search), and "Modificar" (Edit). An error message dialog box is displayed in the foreground, stating: "Error tipos de Datos incorrectos en campo Precio ,Id o Stock" (Error: incorrect data types in field Price, Id or Stock). The dialog box has an "Aceptar" (Accept) button.

Código


```

//////////MODIFICAR//////////

private void iconButton5_Click(object sender, EventArgs e)
{
    try
    {
        conexion.Open();
        if (obtenerDepartamento() != null && !textBox1.Text.Equals("") && !textBox2.Text.Equals("") && !textBox3.Text.Equals("") && !textBox4.Text.Equals(""))
        {
            if (obtenerCampos())
            {
                string cadenamodificar = " UPDATE Inventario set Descripcion = " + Descripcion + " , Precio = " + Precio +
                " , Departamento = " + departamento + " , NumeroDeExistencias= " + stock + " , Imagen= null where Id = " + Id;

                SqlCommand comandoModificar = new SqlCommand(cadenamodificar, conexion);

                DialogResult mensaje = MessageBox.Show("¿Esta Seguro de que quiere modificar el producto de Id " + Id + "?",
                "Seleccione una de las opciones, Sí o No.", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

                if (mensaje == DialogResult.No)
                {
                    MessageBox.Show("La operacion ha sido abortada","Operacion abortada", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                else
                {
                    if (comandoModificar.ExecuteNonQuery() == 1)
                    {
                        MessageBox.Show("Operacion realizada", " Se modificaron los datos del artículo ", MessageBoxButtons.OK, MessageBoxIcon.Information);
                        textBox1.Text = "";
                        textBox2.Text = "";
                        textBox3.Text = "";
                        textBox4.Text = "";
                        comboBox1.SelectedIndex = -1;
                        if (pictureBox2.Image != null)
                        {
                            //evalua si hay imagen en para añadirla y en el caso de que no la hubiera tambien se actualica el campo de la foto a null.
                            SqlCommand ComandoMeterImg = new SqlCommand();
                            ComandoMeterImg.Connection = conexion;
                            ComandoMeterImg.CommandText = "Update Inventario Set Imagen=@Foto where Id=" + Id;
                            ComandoMeterImg.Parameters.Add("@Foto", System.Data.SqlDbType.Image);
                            //Imagen
                            MemoryStream stream = new MemoryStream();
                            pictureBox2.Image.Save(stream, System.Drawing.Imaging.ImageFormat.Png);
                            ComandoMeterImg.Parameters["@Foto"].Value = stream.GetBuffer();
                            ComandoMeterImg.ExecuteNonQuery();
                        }

                        pictureBox2.Image = null;
                    }
                    else
                    {
                        MessageBox.Show(" No existe un artículo con el código introducido ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                        MessageBox.Show(" No existe un artículo con el código introducido ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
            else
            {
                MessageBox.Show("tipos de Datos incorrectos en campo Precio ,Id o Stock", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Asegurese de no dejar los campos vacios.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Asegurese de no dejar los campos vacios.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        conexion.Close();
    }
}

```

Funcionalidad Consultar

El Corte Finlandes

Gestor Base De Datos

Id Producto: 3

Descripción:

Departamento:

Stock:

Imagen:

Precio:

Subir Imagen

Quitar Imagen

Limpiar Datos

Ver Registros

Agregar

Eliminar

Consultar

Modificar

Salir

Simplemente se debe de indicar un Id del producto a consultar, en el caso de que no se sepa dicho Id se puede ver los registros en el botón ver Registros el cual se explicará más adelante.

El Corte Finlandes

Gestor Base De Datos

Id Producto: 3

Descripción: GameBoy

Departamento: Electronica

Stock: 3

Imagen:

Precio: 10

Subir Imagen

Quitar Imagen

Limpiar Datos

Ver Registros

Agregar

Eliminar

Consultar

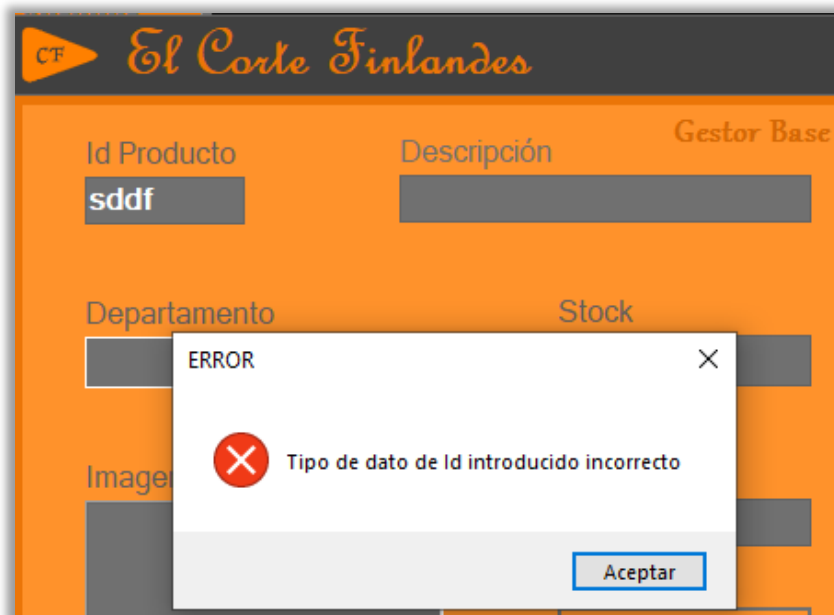
Modificar

Salir

Y posteriormente al pulsar en consultar se rellenarán los datos solos con los del registro en cuestión.

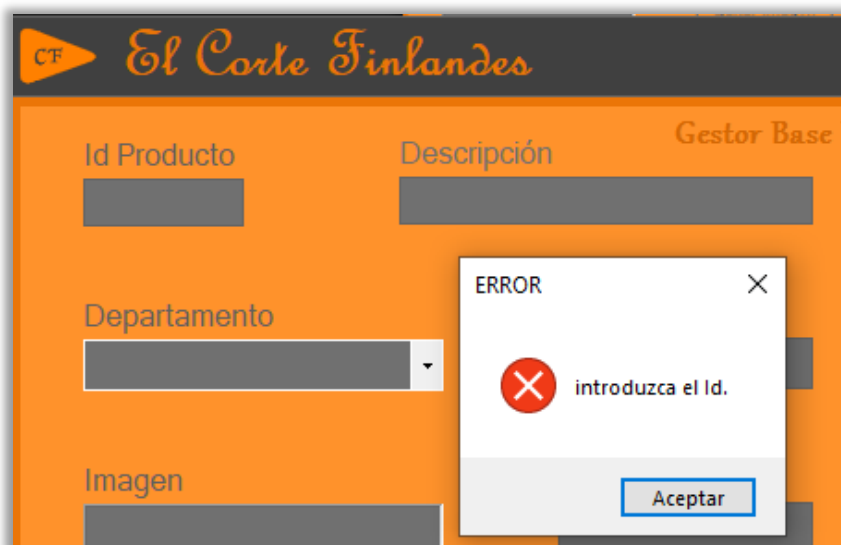
Errores

Consulta de registro de Id de tipo incorrecto



Si introducimos como Id una cadena de texto nos indica que el tipo de datos del Id es incorrecto.

Consulta de registro sin Id



Si no indicamos un Id para consultar nos indicará mediante un messageBox que debemos de introducir un Id para consultar datos.

Consulta de registro Inexistente



Si tratamos de consultar un registro que no se ha creado previamente nos indicará que no existe dicho registro.

Código

```

/////////////////CONSULTAR/////////////////

public void iconButton6_Click(object sender, EventArgs e)
{
    try {
        conexion.Open();
        if (!textBox1.Text.Equals(""))
        {
            //se evalua si hay un Id, si lo hay lo buscara y si no indicará que se debe introducir un Id.
            obtenerCampos();
            int x;
            if (int.TryParse(Id, out x))
            {
                string cadena = "select Descripcion, Precio, Departamento, NumeroDeExistencias from Inventario where Id="
                + textBox1.Text;

                //se crea la cadena que buscara el producto por el Id.
                SqlCommand comando = new SqlCommand(cadena, conexion);
                //se crea un datareader que reciba las filas devueltas por el query.
                SqlDataReader registros = comando.ExecuteReader();
                if (registros.Read())
                {
                    //si se el metodo .Read devuelve true indica que hay al menos un registro y nos hemos desplazado a el
                    //ingresamos el valor de cada columna en su correspondiente textBox.
                    textBox2.Text = registros["Descripcion"].ToString();
                    textBox3.Text = registros["Precio"].ToString();
                    textBox4.Text = registros["NumeroDeExistencias"].ToString();
                    InsertarDepartamento(registros["Departamento"].ToString());

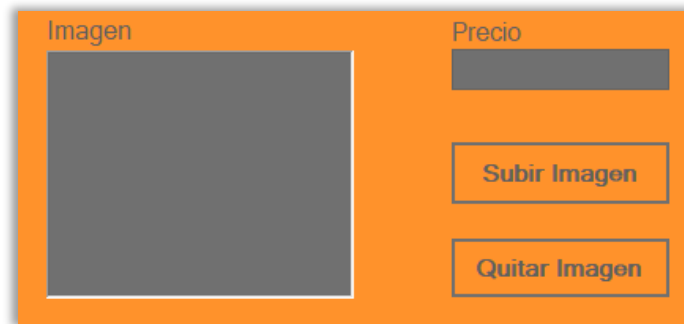
                    //Insercion de imagen.
                    SqlConnection conexionImagen = new SqlConnection("Server = localhost\\SQLEXPRESS; "+
                    "Database = ElCorteFinlandes; integrated security = true");

                    string cadenaImagen = "select Imagen from Inventario where Id=" + textBox1.Text;
                    SqlCommand comandoImagen = new SqlCommand(cadenaImagen, conexionImagen);
                    conexionImagen.Open();
                    SqlDataAdapter da = new SqlDataAdapter(comandoImagen);
                    DataSet ds = new DataSet();
                    da.Fill(ds, "Inventario");//mete la tabla inventario en el dataset ds

                    int c = ds.Tables["Inventario"].Rows.Count;//mira el numero de filas que ha recibido para la tabla inventario
                    if (!Convert.IsDBNull(ds.Tables["Inventario"].Rows[c - 1]["Imagen"]))
                    {
                        //se evalua si se ha obtenido imagen.
                        Byte[] byteImagen = new Byte[0];
                        byteImagen = (Byte[])ds.Tables["Inventario"].Rows[c - 1]["Imagen"];
                        MemoryStream stmBLOBData = new MemoryStream(byteImagen);
                        pictureBox2.Image = Image.FromStream(stmBLOBData);
                    }
                    else
                    {
                        pictureBox2.Image = null;
                    }
                }
                else
                {
                    {
                        MessageBox.Show("No existe un artículo con el código introducido", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
            else
            {
                {
                    MessageBox.Show("Tipo de dato de Id introducido incorrecto", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
        else
        {
            {
                MessageBox.Show("introduzca el Id.", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        conexion.Close();
    }
    catch (Exception ex)
    {
        {
            conexion.Close();
            MessageBox.Show(ex.ToString());
        }
    }
}

```

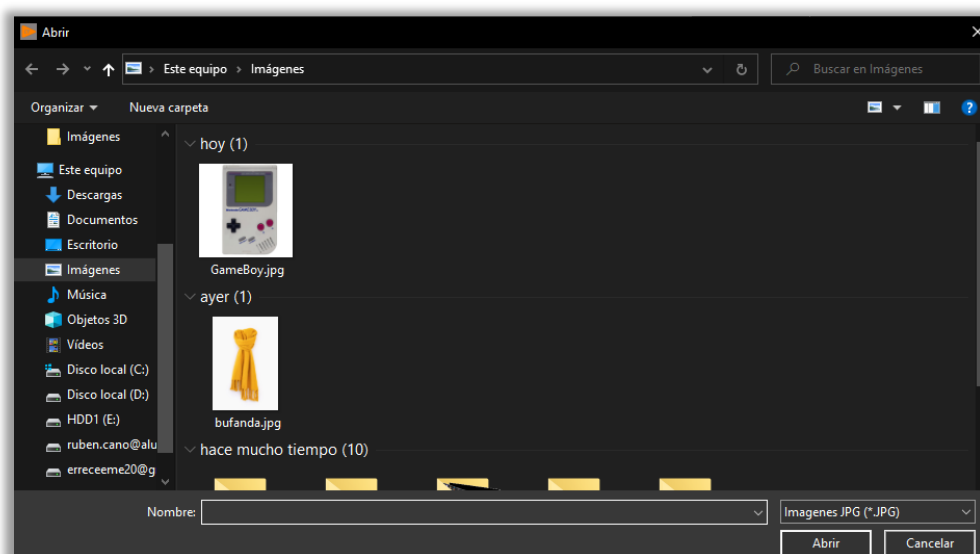
Funcionalidad Imagen



La funcionalidad de la imagen es la forma en la que el usuario puede insertar una imagen para asociarlo a un registro consta de los botones “Subir Imagen” y “Quitar Imagen”

Subir Imagen

Al clicar en subir imagen se abrirá el explorador de archivos en ese momento debe elegir la imagen que desea insertar.



Al seleccionar una imagen esta se agregará al visor de imágenes



Para que el usuario vea lo que va a insertar.

Código

```
private void button2_Click(object sender, EventArgs e)
{
    //Le asignamos un filtro al selector de archivos
    openFileDialog1.Filter = "Imagenes JPG|*.JPG|Imagenes GIF|*.gif|Imagenes Bitmaps|*.bmp|Imagenes PNG|*.png";
    if (openFileDialog1.ShowDialog() == DialogResult.OK) //si el usuario ha hecho clic en el boton de aceptar.
    {
        //asignamos a la imagen que se muestra del pictureBox la imagen que se obtiene del selector de archivos.
        pictureBox2.Image = Image.FromFile(openFileDialog1.FileName);
    }
}
```

Quitar Imagen

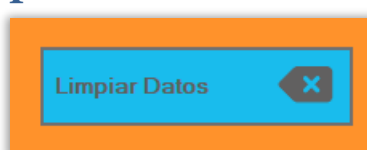


Al clicar en Quitar Imagen el visor volverá a su estado inicial es en este estado que si pulsamos en modificar si están todos los datos bien actualizaremos el registro de forma que en el campo de imagen no habrá una imagen asociada.

Código

```
//////////boton QUITAR Imagen//////////
1 referencia
private void button3_Click(object sender, EventArgs e)
{
    pictureBox2.Image = null; //asignamos la imagen que se muestra como null
}
```

Funcionalidad Limpiar Datos



La funcionalidad de este botón se basa en borrar todos los valores de los campos

Código

```
1 referencia
private void iconButton7_Click(object sender, EventArgs e)//boton para limpiar.
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    comboBox1.SelectedIndex = -1;
    pictureBox2.Image = null;
}
```

Funcionalidad listar registros

Para listar los registros se debe pulsar en el siguiente botón.



Se nos abrirá la siguiente ventana a través de la cual podemos hacer diferentes operaciones.

El Corte Finlandes

Filtrar por: Valor:

	Id	Descripcion	Precio	Departamento	NumeroDeExistencias	Imagen
▶	2	Nintendo DS	55,35	Electronica	2	
	3	GameBoy	10	Electronica	3	
	4	alfa	23	Alimentacion	3	

← Ir atrás

Código

```
1 referencia
private void iconButton9_Click(object sender, EventArgs e)//boton para enseñar todos los registros.
{
    Form Form2= new Form2();
    this.Hide();
    Form2.ShowDialog();
}
```


Filtro

Filtrar por

Id

Id

Descripcion

Precio

Departamento

Numero de Existencia

Primero se debe indicar que campo quieres filtrar

Para posteriormente introducir el valor por el cual quieres filtrar dicho campo.

Valor

3

Y una vez se escriba algo la aplicación enseñara el resultado, es decir, no es necesario un botón de filtrar pues ya lo hace la aplicación automáticamente cuando se escribe.

Filtrar por

Id

Valor

3

	Id	Descripcion	Precio	Departamento	NumeroDeExistencias	Imagen
	3	GameBoy	10	Electronica	3	

Error

Filtrar por

Valor

3

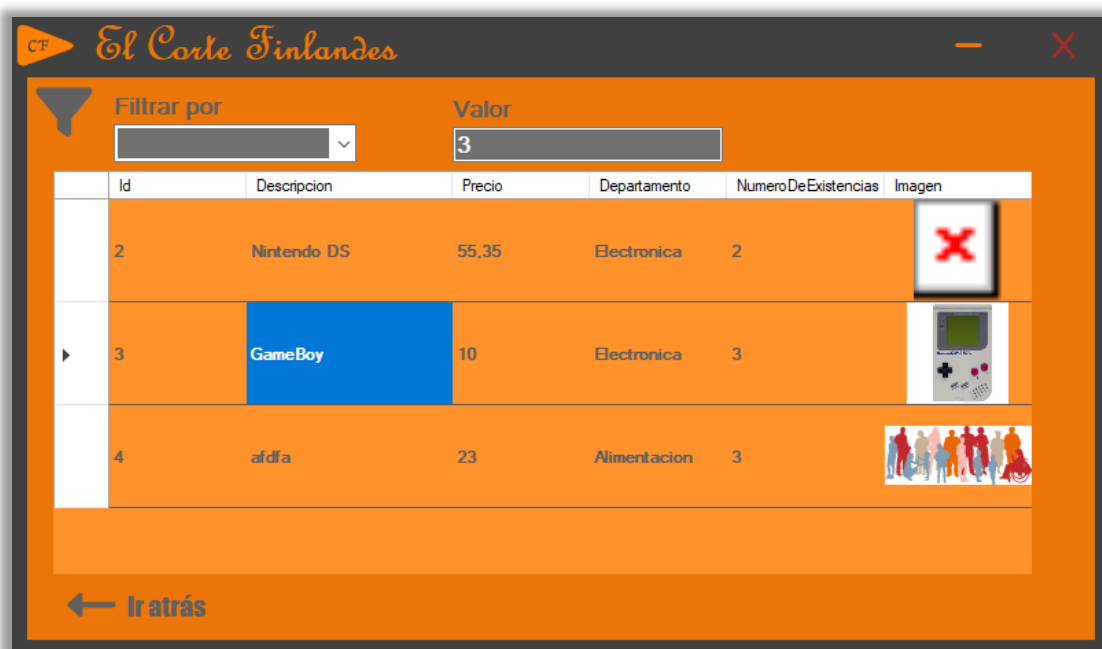
	Id	Descripcion	Precio	Departamento
	2	ERROR		
	3			

Introduzca un campo por el cual quiera filtrar.

Aceptar

Se ha tratado de indicar mediante un messageBox en el caso de que se escriba un valor sin un campo seleccionado que no se está filtrando porque no se ha seleccionado el campo al cual se le quiere aplicar dicho filtro.

Enviar registro a gestor CRUD



Si hacemos doble Clic sobre cualquiera celda o fila se nos abrirá la ventana del gestor CRUD con los datos del registro seleccionado. Por ejemplo, se ha hecho doble click sobre la celda Gameboy, y a continuación esto es lo que se muestra.



Como se puede ver se ha obtenido sus datos en el gestor, como se indicaba anteriormente esto es muy útil a la hora de querer consultar sin

saberse el Id, puesto que lo único que tienes que hacer es filtrar por la descripción o lo que se sepa del registro, por ejemplo se ha filtrado por la Nintendo y el listado de registros nos ha devuelto un registro aunque podría devolvernos más de uno.



Código

```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    System.Windows.Forms.BindingSource fuente = new BindingSource();
    fuente.DataSource = dataGridView2.DataSource;
    if (!textBox1.Text.ToString().Equals(""))
    {
        if (comboBox1.SelectedIndex == -1)
        {
            MessageBox.Show("Introduzca un campo por el cual quiera filtrar.", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            int indiceElegido = comboBox1.SelectedIndex;
            dataGridView2.CurrentCell = null;
            foreach (DataGridViewRow dgvr in dataGridView2.Rows)
            {
                dgvr.Visible = false;
            }
            foreach (DataGridViewRow dgvr in dataGridView2.Rows)
            {
                if (dgvr.Cells[indiceElegido].Value.ToString().ToUpper().Contains(textBox1.Text.ToString().ToUpper()))
                {
                    dgvr.Visible = true;
                }
            }
        }
    }
    else
    {
        dataGridView2.DataSource = fuente;
        //dataGridView2.Update();
    }
}
```

Funcionalidad Ir Atras

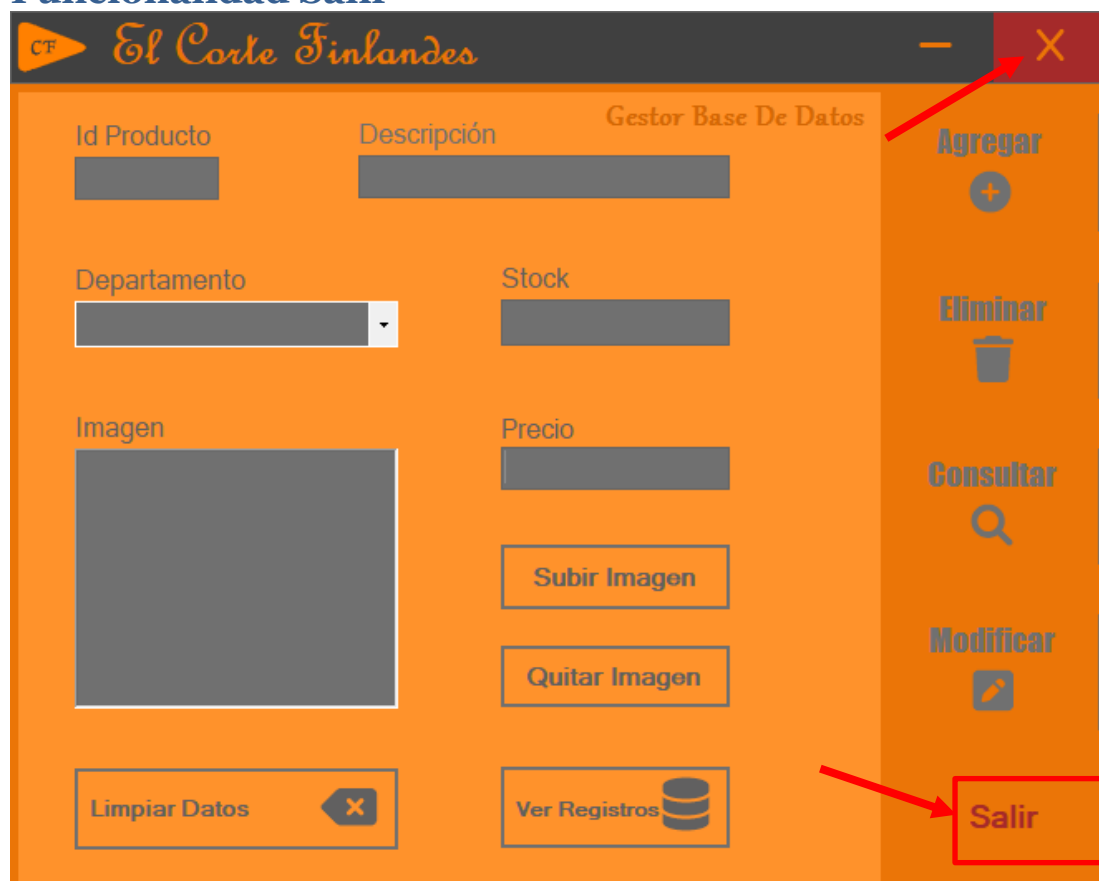
Y finalmente en el caso de que quieras volver a el gestor de CRUD sin seleccionar ningún registro simplemente debes de pulsar en el botón de "Ir atrás".



Código

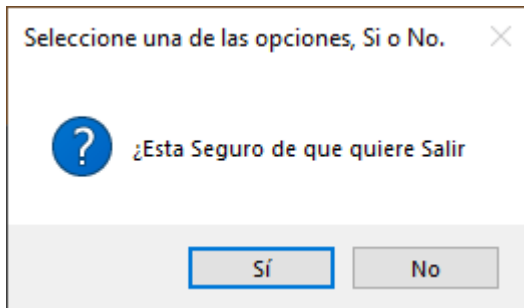
```
1 referencia
private void iconButton1_Click(object sender, EventArgs e)
{
    Form Form1 = new Form1();
    this.Hide();
    Form1.ShowDialog();
}
```

Funcionalidad Salir



Dicha funcionalidad es accesible desde los dos botones señalados en la anterior imagen,

Cuando se pulsa cualquiera de estos dos botones se abre un dialogo en el cual debes confirmar que quieres salir.



Código

```
2 referencias
private void iconButton2_Click(object sender, EventArgs e)//boton de salir.
{
    DialogResult mensaje = MessageBox.Show("¿Esta Seguro de que quiere Salir ",
        "Seleccione una de las opciones, Si o No.", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    if (mensaje != DialogResult.No)
    {
        Application.Exit();
    }
}
```