Manual técnico Proyecto base de datos

Versiones de código utilizadas:

Versión de IDE y código C#: Visual Studio 2022, versión 17.6.

Versión de motor de base de datos SQL Server: SQL Server 2019, versión 15.00.

Requerimientos .NET:

1. Procesador: Se recomienda un procesador con múltiples núcleos y una velocidad de reloj alta. Un procesador de gama media o alta de Intel Core i5 o i7, o su equivalente en AMD, sería una buena opción.

2. Memoria RAM: Se recomienda al menos 8 GB de RAM para un rendimiento óptimo. Sin embargo, si planeas trabajar en proyectos grandes o utilizar varias instancias de Visual Studio al mismo tiempo, es posible que desees considerar 16 GB o más.

3. Almacenamiento: Un disco duro de estado sólido (SSD) es altamente recomendado. Proporciona una mayor velocidad de lectura/escritura en comparación con los discos duros tradicionales, lo que mejora el rendimiento general del sistema y acelera los tiempos de carga de Visual Studio.

4. Tarjeta gráfica: Aunque no es un requisito estricto para Visual Studio, si también planeas utilizar Visual Studio para el desarrollo de aplicaciones gráficas intensivas o juegos, te recomendamos una tarjeta gráfica dedicada de gama media o alta. Esto ayudará en el rendimiento general del sistema y en la aceleración de tareas gráficas.

5. Sistema operativo: Visual Studio es compatible con Windows 10 y versiones posteriores. Asegúrate de tener instalada la versión más reciente de Windows y de mantenerla actualizada.

6. Resolución de pantalla: Se recomienda una resolución de pantalla de al menos 1920x1080 (Full HD) para aprovechar al máximo la interfaz de usuario y la visualización de código en Visual Studio.

7. Conectividad: Asegúrate de tener puertos USB y una conexión a Internet estable para descargar actualizaciones y acceder a recursos en línea.

Recuerda que estos son requisitos recomendados y pueden variar dependiendo de tus necesidades específicas y del tamaño y complejidad de tus proyectos en Visual Studio. Siempre es bueno consultar la documentación oficial de Microsoft para obtener información actualizada sobre los requisitos del sistema para Visual Studio.

Requerimientos SQL Server:

1. Procesador: Se recomienda un procesador de múltiples núcleos y alta velocidad para manejar eficientemente las consultas y cargas de trabajo de SQL Server. Un procesador Intel Core i5 o i7, o su equivalente en AMD, sería una buena opción.

2. Memoria RAM: SQL Server es intensivo en memoria y se recomienda tener al menos 8 GB de RAM. Sin embargo, para bases de datos más grandes o para un mejor rendimiento.

3. Almacenamiento: SQL Server se beneficia de un almacenamiento rápido y confiable. Se recomienda utilizar discos duros de estado sólido (SSD) para el almacenamiento de bases de datos. También es útil tener un disco separado para los archivos de registro de transacciones. Considera utilizar RAID para una mayor protección de datos y rendimiento.

4. Sistema operativo: SQL Server es compatible con varias versiones de Windows Server, así como con algunas versiones de Windows 10. Verifica la documentación oficial de Microsoft para conocer las versiones compatibles.

5. Espacio en disco: Asegúrate de tener suficiente espacio en disco para almacenar tus bases de datos y archivos de registro, así como para las copias de seguridad y los archivos de registro de errores. El espacio requerido variará según el tamaño de tus bases de datos y las necesidades específicas de tu aplicación.

6. Seguridad: Considera implementar medidas de seguridad adecuadas, como contraseñas seguras y autenticación fuerte, para proteger tus bases de datos y prevenir accesos no autorizados.

7. Estos son los requisitos básicos recomendados para ejecutar SQL Server. Ten en cuenta que los requisitos reales pueden variar dependiendo del tamaño y la complejidad de tus bases

de datos, así como de las necesidades específicas de tu aplicación. Consulta la documentación oficial de Microsoft para obtener información actualizada y detallada sobre los requisitos del sistema para SQL Server.

Código fuente con anexo de Forms .NET:

Explorador de soluciones: A continuación, podrá visualizar todos los Windows from utilizados para la creación de la parte gráfica del sistema. Así también todas las conexiones creadas con la base de datos para guardar la información y traerla a la vista del usuario.

Conexión a base de datos: Por medio de las presentes líneas de código establecimos la conexión a la base de datos ubicada en SQLEXPRESS edition.

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Windows.Forms;
using static System.Net.Mime.MediaTypeNames;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using System.IO;
using System.Drawing.Printing;
using System.Net;



namespace proyecto2
{
    public class CConexion
    {
        SqlConnection conex = new SqlConnection();
        string cadena =
"Server=localhost\\SQLEXPRESS;Database=Tarea_baseDatos;Trusted_Connection=True;
";



        //metodo de hacer conexion a la bd
        public SqlConnection establecerConexion()
        {
            try
            {
                conex.ConnectionString = cadena;
                conex.Open();
```

```
            }
            catch (Exception e)
            {
                MessageBox.Show("No se conecto correctamente a la base de
datos, error:" + e.ToString());
            }
            return conex;
        }

        //metodo de cerrar conexion a la bd
        public void cerrarConexion()
        {
            conex.Close();
        }
```

Form Principal: Con el presente Windows from se creó una medida de seguridad para el sistema para que el usuario cuente con un "Usuario" y "Contraseña" los cuales han sido creados desde el código fuente el cual validará si son válidas o no.

Cuenta además con un botón llamado "Ingresar" para validar por medio de condicionales de manera directa por medio de datos quedamos.

Cuenta con dos textbox, uno de ellos nos permite escribir el usuario el cual es "Jonathan" y el segundo para escritura de contraseña "071".

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace proyecto2
{
    public partial class Principal : Form
    {
        public Principal()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {


            if (user.Text == "Jonathan" &&  pasword.Text == "071")
            {
               Opciones ventas = new Opciones();
                ventas.Show();

            }
            else
```

```
            {
                MessageBox.Show("Error al ingresar su 'USUARIO' O
'CONTRASEÑA'");
            }
        }

        private void user_TextChanged(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void Principal_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Form Menu: Este es uno de los forms más importantes, puesto que gracias al presente podemos elegir las una de las dos opciones que deseamos ejecutar en dicho sistema.

Al seleccionar una de las dos opciones lo llevara automáticamente a otra ventana la cual usted decidió ejecutar-

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace proyecto2
{
    public partial class Opciones : Form
    {
        public Opciones()
        {
            InitializeComponent();
        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
```

```
        {

        }

        private void buttonTienda_Click(object sender, EventArgs e)
        {
            Vista_Inventario form1 = new Vista_Inventario();
            form1.Show();

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void COMPRAR_Click(object sender, EventArgs e)
        {
            VENTAS_PRODUCTOS ventas = new VENTAS_PRODUCTOS();
            ventas.Show();
        }

        private void buttonCliente_Click(object sender, EventArgs e)
        {
            Cliente cliente = new Cliente();
            cliente.Show();
        }

        private void Opciones_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Form Inventario_Prodcutos: En esta opción se le mostrara una vista agradable en la cual se le mostrara la vista de la tabla de inventario y a la vez la vista de la tabla de productos y a la vez podrá insertar usted otro producto agrégalo automáticamente a la tabla de inventario y productos.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.ToolTip;

namespace proyecto2
{
    public partial class Vista_Inventario : Form
    {
```

```
        CConexion objetConexion;
        public Vista_Inventario()
        {
            InitializeComponent();

            objetConexion = new CConexion();

        objetConexion.GET_Inventario(DATAGRIDiNVENTARY);
            objetConexion.GET_Productos(dataGridViewProductosMuestra);

        }

        public void BTNAGGPRODUCT_Click(object sender, EventArgs e)
        {
            objetConexion.Products(DESCRIPCIONPRODUCT, IDMARCAPRODUCT,
IDCATEGORIAPRODUCT, PRECIO);
            objetConexion.GET_Inventario(DATAGRIDiNVENTARY);
            objetConexion.GET_Productos(dataGridViewProductosMuestra);


        }

        private void label7_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void STOCK_Click(object sender, EventArgs e)
        {
            AdminStok admin = new AdminStok();
            admin.Show();
        }

        private void dataGridViewProductos_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void label5_Click(object sender, EventArgs e)
        {

        }
    }
}
```

From Stock: es el cual se encuentra un botón llamado “Stock” el cual a la hora de darle clic lo llevara automáticamente al form de stock en cual podrá modificar el Stock para se actualice la vista de las tablas de productos y inventario

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace proyecto2
{
    public partial class AdminStok : Form
    {
        CConexion objetConexion;
        public AdminStok()
        {
            InitializeComponent();
            objetConexion = new CConexion();

        }

        private void AdminStok_Load(object sender, EventArgs e)
        {
            objetConexion.Prodcutos_compra(comboBoxnameproductsADMIN);
        }

        private void AGREGAR_Click(object sender, EventArgs e)
        {

objetConexion.IncrementoStock(TextBoxIdProductoADMIN,CantidadaggADMIN);

        }

        private void comboBoxnameproductsADMIN_SelectedIndexChanged(object
sender, EventArgs e)
        {
            string selectedProduct =
comboBoxnameproductsADMIN.SelectedItem.ToString();
            decimal precio = objetConexion.GetPrecio_Producto(selectedProduct);
            decimal CodigoProducto =
objetConexion.GetI_dProducto(selectedProduct);
            textBoxdelPrecioADMIN.Text = precio.ToString();
            TextBoxIdProductoADMIN.Text = CodigoProducto.ToString();
        }

        private void CantidadaggADMIN_ValueChanged(object sender, EventArgs e)
        {

        }

        private void Stock_Paint(object sender, PaintEventArgs e)
        {
```

```
        }
    }
}
```

Form Ventas: Este es uno de los forms más importantes, puesto que gracias al presente podremos realizar todas las ventas necesarias y poder interactuar y afectar a nuestra base de datos. Fue creada a base de textbox los cuales guardarán información en una tabla llamada "Ventas" en la base de datos maestra.

Cuenta además con un DATAGRIDVIEW el cual mostrará de manera gráfica todos los cambios realizados y a l vez le muestra una vista de los productos disponibles en la base de datos.

Y en ese mismo form puede hacer una compra en cual al llenar los compos que se le muestran en dicho from y darle clic en el botón comprar automaticamente le generara un pdf el cual será la factura.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Printing;


namespace proyecto2
{
    public partial class VENTAS_PRODUCTOS : Form
    {
        CConexion objetConexion;
        public VENTAS_PRODUCTOS()
        {
            InitializeComponent();
            objetConexion = new CConexion();

        }

        PrintDocument printDocument1;

        void Ventas_Load(object sender, EventArgs e)
        {
            objetConexion.GET_Productos(dataGridViewProductosMuestra);
            objetConexion.Prodcutos_compra(comboBoxnameproducts);
            objetConexion.Clientes_compras(comboBoxCLiente);
        }

        void BTNAGGPRODUCT_Click(object sender, EventArgs e)
```

```
        {
            objetConexion.Sales_h(textBoxSeriesSalesH, textBoxIdCliente);
            decimal numeroSalesH = objetConexion.Numero_Factura();
            NumeroSalesH.Text = numeroSalesH.ToString();
            objetConexion.Sales_D(CantidadCompra, TextBoxIdProducto,
NumeroSalesH, textBoxSeriesSalesH);


            printDocument1 = new PrintDocument();
            PrinterSettings settings = new PrinterSettings();


            printDocument1.PrinterSettings = settings;
            printDocument1.PrintPage += CREARFACTURA;

            printDocument1.Print();
        }



        void comboBoxnameproducts_SelectedIndexChanged(object sender, EventArgs
e)
        {
            string selectedProduct =
comboBoxnameproducts.SelectedItem.ToString();
            decimal precio = objetConexion.GetPrecio_Producto(selectedProduct);
            decimal CodigoProducto =
objetConexion.GetI_dProducto(selectedProduct);
            textBoxdelPrecio.Text = precio.ToString();
            TextBoxIdProducto.Text = CodigoProducto.ToString();
        }

        private void comboBoxCLiente_SelectedIndexChanged(object sender,
EventArgs e)
        {
            string selectedClient = comboBoxCLiente.SelectedItem.ToString();
            decimal id = objetConexion.Get_IdCliente(selectedClient);
            textBoxIdCliente.Text = id.ToString();
        }

        private void dataGridViewpruebaclientes_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void textBoxSeriesSalesH_TextChanged(object sender, EventArgs
e)
        {

        }

        private void NumeroSalesH_TextChanged(object sender, EventArgs e)
        {

        }

        public void CREARFACTURA(object sender, PrintPageEventArgs e)
```

```
        {
            Font font = new Font("Arial", 8);
            int ancho = 500;
            int y = 20;

            int Cantidad;
            string cantidacad = CantidadCompra.Text;
            int.TryParse(cantidacad, out Cantidad);

            int Total;
            string text2 = textBoxdelPrecio.Text;
            int.TryParse(text2, out Total);
            Total = Total * Cantidad;

            e.Graphics.DrawString("
_____________________Factura_Modulo_Ventas_____________________ ",
font, Brushes.Black, new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        -
Factura No.:  " + NumeroSalesH.Text + "                              -", font,
Brushes.Black, new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        |------
----------------------------------------------------|", font, Brushes.Black,
new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        -
Cliente:  " + comboBoxCLiente.Text + "                               -", font,
Brushes.Black, new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        |------
----------------------------------------------------|", font, Brushes.Black,
new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        -Fecha:
" + DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss") + "-", font, Brushes.Black,
new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        |------
----------------------------------------------------|", font, Brushes.Black,
new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        -id
- Descripcion         - precio          - cantidad          - total       -",
font, Brushes.Black, new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                        |---|--
----------------------|------------------|----------------------|----------
-----------|", font, Brushes.Black, new RectangleF(0, y += 20, ancho, 20));

            e.Graphics.DrawString("                                          " +
TextBoxIdProducto.Text + "             " + comboBoxnameproducts.Text + "
" + textBoxdelPrecio.Text + "                            " +
cantidacad.ToString() + "             " + Total.ToString() + "
", font, Brushes.Black, new RectangleF(0, y += 20, ancho, 20));
        }
```

```
        private void dataGridViewProductosMuestra_CellContentClick(object
sender, DataGridViewCellEventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Código principal de la clase primordial donde se codifico la funcionalidad del sistema

La clase se llama "CConexion"

```
using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.IO;

using System.Drawing.Printing;

using System.Net;




namespace proyecto2

{

   public class CConexion

   {

      SqlConnection conex = new SqlConnection();

      string cadena =
"Server=localhost\\SQLEXPRESS;Database=Tarea_baseDatos;Trusted_Connection=True;";
```

```
        //metodo de hacer conexion a la bd
        public SqlConnection establecerConexion()
        {
            try
            {
                conex.ConnectionString = cadena;
                conex.Open();
            }
            catch (Exception e)
            {
                MessageBox.Show("No se conecto correctamente a la base de datos, error:" +
e.ToString());
            }
            return conex;
        }

        //metodo de cerrar conexion a la bd
        public void cerrarConexion()
        {
            conex.Close();
        }

        //metodo para insertar datos en la tabla products
        public void Products(System.Windows.Forms.TextBox Descripcion, NumericUpDown id_Marca,
NumericUpDown id_Categoria, NumericUpDown precio)
        {
            try
            {
```

```
            String query = "INSERT INTO products (Descripcion, id_Marca, id_Categoria, precio,
Imagen) values('" + Descripcion.Text.ToString() + "'," + id_Marca.Value + "," + id_Categoria.Value +
"," + precio.Value + "," + "'Imagen');";
            SqlCommand cmd = new SqlCommand(query, establecerConexion());
            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Se inserto correctamente el producto ");
            cerrarConexion();
        }
        catch (Exception ex)
        {
            MessageBox.Show("No se pudo insertar el registro" + ex);
        }
    }


    //metodo para mostrar el inventario a un datagrid
    public void GET_Inventario(DataGridView DATAGRIDiNVENTORY)
    {
        try
        {
            DATAGRIDiNVENTORY.DataSource = null;
            SqlDataAdapter adapter = new SqlDataAdapter("CargarInventario", establecerConexion());
            adapter.SelectCommand.CommandType = CommandType.StoredProcedure;
            DataTable dt = new DataTable();
            adapter.Fill(dt);
            DATAGRIDiNVENTORY.DataSource = dt;
            cerrarConexion();
        }
        catch (Exception ex)
        {
```

```
        MessageBox.Show("No se logro cargar los registros al inventario : " + ex);
    }
}


//metodo para cargar los productos a un datagrid
public void GET_Productos(DataGridView dataGridViewProductos)
{
    try
    {
        dataGridViewProductos.DataSource = null;
        SqlDataAdapter adapter = new SqlDataAdapter("CargarProducto", establecerConexion());
        adapter.SelectCommand.CommandType = CommandType.StoredProcedure;
        DataTable dt = new DataTable();
        adapter.Fill(dt);
        dataGridViewProductos.DataSource = dt;
        cerrarConexion();
    }
    catch (Exception ex)
    {
        MessageBox.Show("No se logro cargar los registros al productos : " + ex);
    }
}



//metodo para cargar los cliente a un combobox
public void Clientes_compras(System.Windows.Forms.ComboBox comboBoxCLiente)
{
```

```
    string query = "SELECT Nombre FROM customer";
    SqlConnection connection = establecerConexion();
    SqlCommand command = new SqlCommand(query, connection);
    SqlDataReader reader = command.ExecuteReader();

    while (reader.Read())
    {
        string nombreCliente = reader["Nombre"].ToString();
        comboBoxCLiente.Items.Add(nombreCliente);
    }
    reader.Close();
    connection.Close();
}

//metodo para cargar el precio del producto a la vez que se selecciona en el combo box
public decimal Get_IdCliente(string nombreCliente)
{
    decimal idClient = 0;
    string query = "SELECT CodigoCliente FROM customer WHERE Nombre = @nombre";
    using (SqlConnection connection = establecerConexion())
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@nombre", nombreCliente);
            object result = command.ExecuteScalar();
            if (result != null && result != DBNull.Value)
            {
                idClient = Convert.ToDecimal(result);
            }
```

```
        }
    }
    return idClient;
}


//metodo para cargar los productos a un combobox
public void Prodcutos_compra(System.Windows.Forms.ComboBox ComboBoxnameproducts)
{
    string query = "SELECT Descripcion FROM products";
    SqlConnection connection = establecerConexion();
    SqlCommand command = new SqlCommand(query, connection);
    SqlDataReader reader = command.ExecuteReader();

    while (reader.Read())
    {
        string nombreProducto = reader["Descripcion"].ToString();
        ComboBoxnameproducts.Items.Add(nombreProducto);
    }
    reader.Close();
    connection.Close();

}


//metodo para cargar el precio del producto a la vez que se selecciona en el combo box
public decimal GetPrecio_Producto(string nombreProducto)
{
    decimal precio = 0;
```

```
    string query = "SELECT Precio FROM products WHERE Descripcion = @nombre";
    using (SqlConnection connection = establecerConexion())
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@nombre", nombreProducto);
            object result = command.ExecuteScalar();
            if (result != null && result != DBNull.Value)
            {
                precio = Convert.ToDecimal(result);
            }
        }
    }
    return precio;
}


//metodo para cargar a un textbox el id del producto al seleccionarlo en el combobox
public decimal GetI_dProducto(string idProducto)
{
    decimal id = 0;
    string query = "SELECT idProducto FROM products WHERE Descripcion = @nombre";
    using (SqlConnection connection = establecerConexion())
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@nombre", idProducto);
            object result = command.ExecuteScalar();
            if (result != null && result != DBNull.Value)
```

```
            {
                id = Convert.ToDecimal(result);
            }
        }
    }
    return id;
}

//metodo para insertar los productos a un combobox
public int Numero_Factura()
{
    string query = "SELECT MAX(Numero) FROM sales_h";
    SqlConnection connection = establecerConexion();
    SqlCommand command = new SqlCommand(query, connection);
    object reader = command.ExecuteScalar();
      int lastNumero = 0;
    if (reader != null && reader != DBNull.Value)
    {
     lastNumero = Convert.ToInt32(reader);

    }
    connection.Close();
    return lastNumero;
}
```

```
//metodo para cargar los datos de la tabla sales_d


public void Sales_D(NumericUpDown CantidadCompra, System.Windows.Forms.TextBox
TextBoxIdProducto, System.Windows.Forms.TextBox NumeroSalesH,
System.Windows.Forms.NumericUpDown textBoxSeriesSalesH)
{
    int idProductoN;
    string text = TextBoxIdProducto.Text;
    int.TryParse(text, out idProductoN);


    int NumeroFactura;
    string text2 = NumeroSalesH.Text;
    int.TryParse(text2, out NumeroFactura);
    try
    {
        String query = "INSERT INTO sales_d (Cantidad, Articulo, Factura, Serie) VALUES(" +
CantidadCompra.Value + "," + idProductoN + "," + NumeroFactura + ", " +
textBoxSeriesSalesH.Value + ");";
        SqlCommand cmd = new SqlCommand(query, establecerConexion());
        SqlDataReader dr = cmd.ExecuteReader();
        MessageBox.Show("Se inserto correctamente la venta ");
        cerrarConexion();




        string querydos = "UPDATE inventory SET stock_in = stock_in-" + CantidadCompra.Value +
", outlets = outlets+"+ CantidadCompra.Value + " WHERE id_product = " + idProductoN + ";";
```

```
            SqlCommand cmdd = new SqlCommand(querydos, establecerConexion());
            SqlDataReader drd = cmdd.ExecuteReader();
            MessageBox.Show("Se modifico el stock ");
            cerrarConexion();
        }
        catch (Exception ex)
        {
            MessageBox.Show("No se pudo insertar el registro" + ex);
        }
    }

    public void IncrementoStock(System.Windows.Forms.TextBox TextBoxIdProducto,
System.Windows.Forms.NumericUpDown Cantidadagg)
    {
        int idprodudctoentero = Convert.ToInt32(TextBoxIdProducto.Text);
        try
        {
            MessageBox.Show("UPDATE inventory SET stock_in = stock_in+" + Cantidadagg.Value + ",
entries = entries +" + Cantidadagg.Value + " WHERE id_product = " + idprodudctoentero + ";");
            string selectQuery = "UPDATE inventory SET stock_in = stock_in+" + Cantidadagg.Value + ",
entries = entries +"+ Cantidadagg.Value+" WHERE id_product = " + idprodudctoentero + ";";
            SqlCommand cmd = new SqlCommand(selectQuery, establecerConexion());
            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Se actualizo correctamente el Stock");
            cerrarConexion();
        }
        catch
        {
```

```
        MessageBox.Show("No actualizo el Stock");
    }
}




//Metodo para visualizar la factura en Html



//metodo para insetar el nombre a un combobox
public void Clientes_ventas(System.Windows.Forms.ComboBox comboBoxCLiente)
{
    string query = "SELECT Nombre FROM customer";
    SqlConnection connection = establecerConexion();
    SqlCommand command = new SqlCommand(query, connection);
    SqlDataReader reader = command.ExecuteReader();

    while (reader.Read())
    {
        string nombreCliente = reader["Nombre"].ToString();
        comboBoxCLiente.Items.Add(nombreCliente);
    }
    reader.Close();
    connection.Close();
}

// metodo para cargar los datos de la tabla customer
```

```
    public void Clientes(System.Windows.Forms.TextBox NombreCliente,
System.Windows.Forms.TextBox DireccionCliente, System.Windows.Forms.TextBox CorreoCliente,
System.Windows.Forms.TextBox NitCliente)
    {
        try
        {
            String query = "INSERT INTO customer (Nombre, Direccion, Correo, Nit) VALUES ('" +
NombreCliente.Text.ToString() + "','" + DireccionCliente.Text.ToString() + "','" +
CorreoCliente.Text.ToString() + "','" + NitCliente.Text.ToString() + "',);";
            SqlCommand cmd = new SqlCommand(query, establecerConexion());
            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Se inserto correctamente el producto ");
            cerrarConexion();
        }
        catch (Exception ex)
        {
            MessageBox.Show("No se pudo insertar el registro" + ex);
        }
    }


    //metodo para cargar los datos de la tabla sales_h


    public void Sales_h(NumericUpDown textBoxSeriesSalesH, System.Windows.Forms.TextBox
textBoxIdCliente)
    {
        int numero;
        string text = textBoxIdCliente.Text;
        int.TryParse(text, out numero);
        try
        {
```

```
            String query = "INSERT INTO sales_h (Serie, Fecha, Codigo_Cliente, Nit) VALUES(" +
textBoxSeriesSalesH.Value + ",'" + DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss") + "'," +
numero + ", 1);";

            SqlCommand cmd = new SqlCommand(query, establecerConexion());

            SqlDataReader dr = cmd.ExecuteReader();

            MessageBox.Show("Se inserto correctamente el producto ");

            cerrarConexion();

        }

        catch (Exception ex)

        {

            MessageBox.Show("No se pudo insertar el registro" + ex);

        }

    }


  }

}
```

Script Base de Datos Proyecto:

Explicación de la función de la base de datos: La importancia para la conexión con la solución en .Net es crucial, ya que este va a ser el almacén de nuestros sistemas con el cual poblaremos nuestra base de datos desde cero.

Por otro lado, vamos a contar con tablas que serán utilizadas con diferente propósito.

```
CREATE DATABASE  Tarea_baseDatos


USE Tarea_baseDatos



CREATE TABLE customer (
CodigoCliente INT IDENTITY(1,1) PRIMARY KEY,
Nombre VARCHAR(50),
Direccion VARCHAR(50),
Correo VARCHAR(50),
Nit VARCHAR(25)
);

CREATE TABLE company(
Nit INT IDENTITY (1,1) PRIMARY KEY,
Nombre VARCHAR(50),
Direccion VARCHAR(50),
Correo VARCHAR(50),
Telefono VARCHAR(25),
Portal VARCHAR(25)
);

CREATE TABLE products(
idProducto INT IDENTITY(1,1) PRIMARY KEY,
Descripcion VARCHAR(50),
id_Marca INT,
id_Categoria INT,
precio INT,
Imagen VARCHAR(max)
);

CREATE TABLE sales_h(
Numero INT IDENTITY(1,1) PRIMARY KEY,
Serie INT,
Fecha DATETIME,
Codigo_Cliente INT,
Nit INT,
FOREIGN KEY (Codigo_Cliente) REFERENCES customer (CodigoCliente),
FOREIGN KEY (Nit) REFERENCES company (Nit)
);
```

```sql
CREATE TABLE sales_d(
Id INT IDENTITY(1,1) PRIMARY KEY,
Cantidad INT,
Articulo INT,
Factura INT,
Serie INT
FOREIGN KEY (Articulo) REFERENCES products (idProducto),
FOREIGN KEY (Factura) REFERENCES sales_h (Numero),
);

CREATE TABLE inventory(
id_inventory INT IDENTITY(1,1) PRIMARY KEY,
date_int DATE not null,
date_out DATE not null,
id_product INT not null,
stock_in INT not null,
entries INT not null,
outlets INT not null,
FOREIGN KEY (id_product) REFERENCES products (IdProducto),
);



INSERT INTO customer (Nombre, Direccion, Correo, Nit) VALUES
('Luis', 'Calle 1', 'Luis.com', '1234567890'),
('karen', 'Calle 2', 'karen.com', '0987654321'),
('Damaris', 'Calle 3', 'Damaris.com', '2468101214');

INSERT INTO company (Nombre, Direccion, Correo, Telefono, Portal) VALUES
('Gallo', 'Calle 1', 'Gallo.com', '12345678', 'www.uno.com'),
('Pepsi', 'Calle 2', 'Pepsi.com', '87654321', 'www.dos.com'),
('Bigcola', 'Calle 3', 'Bigcola.com', '13579246', 'www.tres.com');

INSERT INTO products (Descripcion, id_Marca, id_Categoria, precio, Imagen)
VALUES ('Gallo', 1, 1, 10, 'imagen1.jpg'),
       ('Montecarlo', 1, 2, 20, 'imagen2.jpg'),
       ('Heineken', 2, 1, 15, 'imagen3.jpg'),
       ('Frijol', 2, 2, 25, 'imagen4.jpg'),
       ('Juegos', 3, 1, 12, 'imagen5.jpg');


INSERT INTO sales_h (Serie, Fecha, Codigo_Cliente, Nit) VALUES
(001, '2023-05-12', 1, 1),
(001, '2023-05-12', 2, 1),
(001, '2023-05-12', 3, 1);

INSERT INTO sales_h (Serie, Fecha, Codigo_Cliente, Nit) VALUES
(002, '2023-05-12', 1, 1);

INSERT INTO sales_d (Cantidad, Articulo, Factura, Serie) VALUES
(1, 1, 2, 001);

select * from sales_h
select * from sales_d
```

```sql
INSERT INTO inventory (date_int, date_out, id_product, stock_in, entries, outlets)
VALUES
('2023-05-01', '2023-05-12', 1, 50, 20, 10);

CREATE TRIGGER insertarInventario
ON products
AFTER INSERT
AS
BEGIN
INSERT INTO inventory
(date_int, date_out, id_product, stock_in, entries, outlets)
SELECT GETDATE(),GETDATE(), idProducto,1,1,0 FROM inserted
END


SELECT id_inventory, date_int, date_out, p.Descripcion , stock_in, entries, outlets
 FROM inventory AS iv
 inner join products AS p ON iv.id_product = p.idProducto

 select stock_in from inventory where id_product = 1;
 update inventory Set stock_in=stock_in+1 where id_product = 1;

 CREATE PROCEDURE CargarInventario
AS
 SELECT id_inventory, date_int, date_out, p.Descripcion , stock_in, entries, outlets
 FROM inventory AS iv
 inner join products AS p ON iv.id_product = p.idProducto
GO

EXEC CargarInventario

 CREATE PROCEDURE CargarProducto
AS
 SELECT idProducto, Descripcion, id_Marca,id_Categoria, precio,Imagen
 FROM products
GO
EXEC CargarProducto

 CREATE PROCEDURE CargarClientes
AS
 SELECT CodigoCliente, Nombre, Direccion, Correo, Nit
 FROM customer
GO
EXEC CargarClientes

select idProducto from products
```