

```

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 VentaEntidades;
using VentaNegocio;

namespace VentaPresentacion
{
    public partial class Form_Venta : Form
    {
        ClienteEntidad cliente = null;
        DataGridViewComboBoxEditingControl dgvCombo;

        public Form_Venta()
        {
            InitializeComponent();
        }

        private void button_BuscarCliente_Click(object sender, EventArgs e)
        {
            Form_Cliente frm = new Form_Cliente();
            if (frm.ShowDialog() == DialogResult.OK)
            {
                cliente = ClienteNegocio.DevolverClientePorId(frm.IdCliente);
                textBox_Id.Text = cliente.Id.ToString();
                textBox_Nombre.Text = cliente.Nombre;
                textBox_Apellido.Text = cliente.Apellido;
                textBox_Cedula.Text = cliente.Cedula;
                textBox_Direccion.Text = cliente.Direccion;
                textBox_Telefono.Text = cliente.Telefono;
                textBox_Celular.Text = cliente.Celular;
            }
        }

        private void Form_Venta_Load(object sender, EventArgs e)
        {
            CargarComboProductos();
            NuevaLinea();
        }

        private void NuevaLinea()
        {
            List<DGVLine> newlist = new List<DGVLine>();
            if (dataGridView_Productos.DataSource != null)
            {
                // List<DGVLine> list = (dataGridView_Productos.DataSource as List<DGVLine>);
                List<DGVLine> list = (List<DGVLine>)dataGridView_Productos.DataSource;
                list.ForEach(item =>
                {
                    newlist.Add((DGVLine)item.Clone());
                });
            }
            newlist.Add(new DGVLine());

            dataGridView_Productos.AutoGenerateColumns = false;
            dataGridView_Productos.DataSource = newlist;
        }
    }
}

```

```

    }

    private void CargarComboProductos()
    {
        DataGridViewComboBoxColumn comboCol = dataGridView_Productos.Columns["producto"] as
DataGridViewComboBoxColumn;
        comboCol.ValueMember = "id";
        comboCol.DisplayMember = "nombre";
        comboCol.DataSource = ProductoNegocio.DevolverListaProductos();
    }

    private void button_AgregarProducto_Click(object sender, EventArgs e)
    {
        NuevaLinea();
    }

    private void dataGridView_Productos_CellEndEdit(object sender, DataGridViewCellEventArgs e)
    {
        if (dgvCombo!=null)
        {
            dgvCombo.SelectedIndexChanged -= DgvCombo_SelectedIndexChanged;
        }
    }

    private void dataGridView_Productos_EditingControlShowing(object sender,
DataGridViewEditingControlShowingEventArgs e)
    {
        dgvCombo = e.Control as DataGridViewComboBoxEditingControl;

        if (dgvCombo != null)
        {
            dgvCombo.SelectedIndexChanged += DgvCombo_SelectedIndexChanged;
        }
    }

    private void DgvCombo_SelectedIndexChanged(object sender, EventArgs e)
    {
        ComboBox combo = sender as ComboBox;
        DataGridViewRow row = dataGridView_Productos.CurrentRow;
        row.Cells["PrecioUnitario"].Value =
ProductoNegocio.DevolverPrecioPorId(Convert.ToInt32(combo.SelectedValue));
    }

    bool Validaciones()
    {
        errorProvider1.Clear();
        bool resultado = true;
        if (string.IsNullOrEmpty(textBox_Nombre.Text))
        {
            errorProvider1.SetError(textBox_Nombre, "El nombre es obligatorio");
            resultado = false;
        }

        if (string.IsNullOrEmpty(textBox_Cedula.Text))
        {
            errorProvider1.SetError(textBox_Cedula, "La cedula es obligatoria");
            resultado = false;
        }

        return resultado;
    }

```

```

private void button_ConfirmarVenta_Click(object sender, EventArgs e)
{
    if (!Validaciones())
    {
        return;
    }
    if (cliente==null)
    {
        cliente = new ClienteEntidad();
    }

    if (cliente != null)
    {
        cliente.Nombre = textBox_Nombre.Text;
        cliente.Apellido = textBox_Apellido.Text;
        cliente.Cedula = textBox_Cedula.Text;
        cliente.Direccion = textBox_Direccion.Text;
        cliente.Telefono = textBox_Telefono.Text;
        cliente.Celular = textBox_Celular.Text;

        cliente = ClienteNegocio.Guardar(cliente);
    }

    VentasEntidades venta = new VentasEntidades();
    venta.IdCliente = cliente.Id;
    venta.FechaVenta = DateTime.Now.Date;

    venta.listaDetalle = new List<VentaDetalleEntidades>();

    foreach (DataGridViewRow row in dataGridView_Productos.Rows)
    {
        VentaDetalleEntidades detalleVenta = new VentaDetalleEntidades();
        detalleVenta.IdProducto = Convert.ToInt32(row.Cells["Producto"].Value);
        detalleVenta.PrecioUnitario = Convert.ToInt32(row.Cells["PrecioUnitario"].Value);
        detalleVenta.Cantidad = Convert.ToInt32(row.Cells["Cantidad"].Value);
        venta.listaDetalle.Add(detalleVenta);
    }
    VentasNegocio.RegistrarVenta(venta);
}

```

```

public class DGVLine : ICloneable
{
    public int? Producto { get; set; }
    public decimal PrecioUnitario { get; set; }
    public int Cantidad { get; set; }

    #region ICloneable Members

    public object Clone()
    {
        DGVLine item = new DGVLine();

        item.Producto = this.Producto;
        item.PrecioUnitario = this.PrecioUnitario;
        item.Cantidad = this.Cantidad;

        return item;
    }

    #endregion
}

```

```

}

```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace VentaEntidades
{
    public class VentasEntidades
    {
        public int Id { get; set; }
        public int IdCliente { get; set; }
        public DateTime FechaVenta { get; set; }
        public List<VentaDetalleEntidades> listaDetalle { get; set; }
        public decimal Total
        {
            get { return this.listaDetalle.Sum(x => x.PrecioUnitario * x.Cantidad); }
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VentaEntidades;
using VentaDatos;
namespace VentaNegocio
{
    public class ClienteNegocio
    {
        public static List<ClienteEntidad> DevolverListaClientes()
        {
            return ClienteDatos.DevolverListaClientes();
        }

        public static ClienteEntidad DevolverClientePorId(int idCliente)
        {
            return ClienteDatos.DevolverClientePorId(idCliente);
        }

        public static ClienteEntidad Guardar(ClienteEntidad cliente)
        {
            if (cliente.Id!=0)
            {
                return ClienteDatos.Actualizar(cliente);
            }
            else
            {
                return ClienteDatos.Nuevo(cliente);
            }
        }
    }
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VentaEntidades;
using VentaDatos;
using System.Transactions;

namespace VentaNegocio
{
    public static class VentasNegocio
    {
        public static void RegistrarVenta(VentasEntidades ventas)
        {
            using (TransactionScope scope = new TransactionScope())
            {
                VentasDatos.Crear(ventas);
                VentasDatos.UpdateTotal(ventas.Id, ventas.Total);
                scope.Complete();
            }
        }
    }
}
```

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VentaEntidades;

namespace VentaDatos
{
    public class ClienteDatos
    {
        public static List<ClienteEntidad> DevolverListaClientes()
        {
            List<ClienteEntidad> lista = new List<ClienteEntidad>();
            using (SqlConnection cn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
            {
                cn.Open();
                string sql = @"SELECT[id]
                                ,[nombre]
                                ,[apellido]
                                ,[cedula]
                                ,[direccion]
                                ,[telefono]
                                ,[celular]
                                FROM[dbo].[Cliente]
                                ORDER BY [apellido]";
                SqlCommand cmd = new SqlCommand(sql, cn);
                SqlDataReader reader = cmd.ExecuteReader();
                while (reader.Read())
                {
                    lista.Add(CargarCliente(reader));
                }
            }
            return lista;
        }

        public static ClienteEntidad Nuevo(ClienteEntidad cliente)
        {
            using (SqlConnection conn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
            {
                conn.Open();
                string sql = @"INSERT INTO [dbo].[Cliente]
                                ([nombre]
                                ,[apellido]
                                ,[cedula]
                                ,[direccion]
                                ,[telefono]
                                ,[celular])
                                VALUES (@nombre, @apellido, @cedula, @direccion, @telefono,@celular)
                                SELECT SCOPE_IDENTITY()";

                SqlCommand cmd = new SqlCommand(sql, conn);

                cmd.Parameters.AddWithValue("@nombre", cliente.Nombre);
                cmd.Parameters.AddWithValue("@apellido", cliente.Apellido);
                cmd.Parameters.AddWithValue("@cedula", cliente.Cedula);
                cmd.Parameters.AddWithValue("@direccion", cliente.Direccion);
                cmd.Parameters.AddWithValue("@telefono", cliente.Telefono);
                cmd.Parameters.AddWithValue("@celular", cliente.Celular);
            }
        }
    }
}

```

```

        cliente.Id = Convert.ToInt32(cmd.ExecuteScalar());
    }

    return cliente;
}

public static ClienteEntidad Actualizar(ClienteEntidad cliente)
{
    using (SqlConnection conn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
    {
        conn.Open();

        string sql = @"UPDATE Cliente
                        SET nombre = @nombre
                        ,apellido = @apellido
                        ,cedula = @cedula
                        ,direccion = @direccion
                        ,telefono = @telefono
                        ,celular = @celular
                        WHERE id= @id";

        SqlCommand cmd = new SqlCommand(sql, conn);

        cmd.Parameters.AddWithValue("@nombre", cliente.Nombre);
        cmd.Parameters.AddWithValue("@apellido", cliente.Apellido);
        cmd.Parameters.AddWithValue("@cedula", cliente.Cedula);
        cmd.Parameters.AddWithValue("@direccion", cliente.Direccion);
        cmd.Parameters.AddWithValue("@telefono", cliente.Telefono);
        cmd.Parameters.AddWithValue("@celular", cliente.Celular);
        cmd.Parameters.AddWithValue("@id", cliente.Id);
        cmd.ExecuteNonQuery();
    }
    return cliente;
}

public static ClienteEntidad DevolverClientePorId(int idCliente)
{
    ClienteEntidad cliente = new ClienteEntidad();
    using (SqlConnection cn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
    {
        cn.Open();
        string sql = @"SELECT[id]
                        ,[nombre]
                        ,[apellido]
                        ,[cedula]
                        ,[direccion]
                        ,[telefono]
                        ,[celular]
                        FROM[dbo].[Cliente]
                        WHERE [id] = @clienteId";

        SqlCommand cmd = new SqlCommand(sql, cn);
        cmd.Parameters.AddWithValue("@clienteId", idCliente);
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            cliente = CargarCliente(reader);
        }
    }
    return cliente;
}

public static ClienteEntidad CargarCliente(IDataReader reader)
{
    ClienteEntidad cliente = new ClienteEntidad();

```



```

        cliente.Id = Convert.ToInt32(reader["id"]);
        cliente.Nombre = Convert.ToString(reader["nombre"]);
        cliente.Apellido = Convert.ToString(reader["apellido"]);
        cliente.Cedula = Convert.ToString(reader["cedula"]);
        cliente.Direccion = Convert.ToString(reader["direccion"]);
        cliente.Telefono = Convert.ToString(reader["telefono"]);
        cliente.Celular = Convert.ToString(reader["celular"]);
        return cliente;
    }
}
}

```

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VentaEntidades;

namespace VentaDatos
{
    public static class VentasDatos
    {
        public static void Crear(VentasEntidades ventas)
        {
            using (SqlConnection conn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
            {
                conn.Open();
                string sqlFactura = @"INSERT INTO [dbo].[Venta]
                                    ([id_Cliente]
                                    ,[fechaVenta]
                                    ,[Total])
                                    VALUES (@idCliente, @fechaVenta, @total)
                                    SELECT SCOPE_IDENTITY()";

                using (SqlCommand cmd = new SqlCommand(sqlFactura, conn))
                {
                    cmd.Parameters.AddWithValue("@idCliente", ventas.IdCliente);
                    cmd.Parameters.AddWithValue("@fechaVenta", ventas.FechaVenta);
                    cmd.Parameters.AddWithValue("@total", 0);
                    ventas.Id = Convert.ToInt32(cmd.ExecuteScalar());
                }

                string sqlDetalle = @"INSERT INTO [dbo].[VentaDetalle]
                                    ([id_Venta]
                                    ,[id_Producto]
                                    ,[precioUnitario]
                                    ,[cantidad])
                                    VALUES (@idVenta, @idProducto, @precioUnitario, @cantidad)
                                    SELECT SCOPE_IDENTITY()";

                using (SqlCommand cmd = new SqlCommand(sqlDetalle, conn))
                {
                    foreach (VentaDetalleEntidades detalle in ventas.listaDetalle)

```

```

        {
            //
            // como se reutiliza el mismo objeto SqlCommand es necesario limpiar los
parametros
            // de la operacion previa, sino estos se iran agregando en la coleccion,
generando un fallo
            //
            cmd.Parameters.Clear();

            cmd.Parameters.AddWithValue("@idVenta", ventas.Id);
            cmd.Parameters.AddWithValue("@idProducto", detalle.IdProducto);
            cmd.Parameters.AddWithValue("@precioUnitario", detalle.PrecioUnitario);
            cmd.Parameters.AddWithValue("@cantidad", detalle.Cantidad);

            detalle.Id = Convert.ToInt32(cmd.ExecuteScalar());
        }
    }
}

public static void UpdateTotal(int idVenta, decimal total)
{
    using (SqlConnection conn = new SqlConnection(ConfiguracionApp.Default.ConexionVentasSql))
    {
        conn.Open();

        string sqlUpdateTotal = @"UPDATE Venta SET Total = @total WHERE id = @ventaId";

        using (SqlCommand cmd = new SqlCommand(sqlUpdateTotal, conn))
        {
            cmd.Parameters.AddWithValue("@total", total);
            cmd.Parameters.AddWithValue("@ventaId", idVenta);

            cmd.ExecuteNonQuery();
        }
    }
}
}
}
}
}

```