



Manual técnico

REQUERIMIENTOS DE DESARROLLO

VISUAL STUDIO

Visual Studio es un entorno
de desarrollo integrado

(IDE) para windows y macOS.

Es compatible con multiples lenguajes
de programación, esta programa permite
crear sitios y aplicaciones web
entorno a la plataforma .NET.



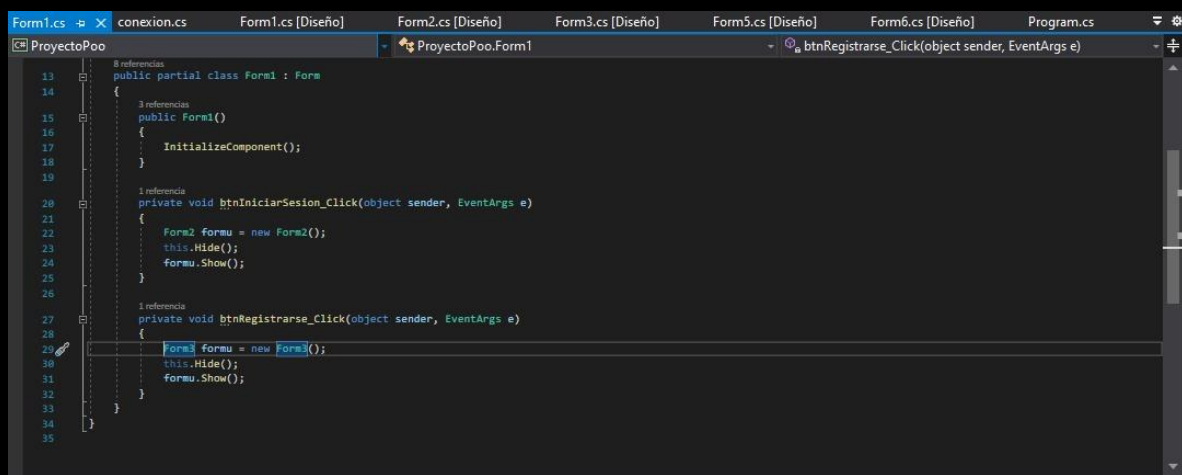
SQL Server Management Studio

se utiliza para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server. La herramienta incluye tanto editores de secuencias de comandos como herramientas gráficas que funcionan con objetos y funciones del servidor.



ESTRUCTURA DEL PROGRAMA.

Inicio.



```
13 3 referencias
14 public partial class Form1 : Form
15 {
16     3 referencias
17     public Form1()
18     {
19         InitializeComponent();
20     }
21
22     1 referencia
23     private void btnIniciarSesion_Click(object sender, EventArgs e)
24     {
25         Form2 formu = new Form2();
26         this.Hide();
27         formu.Show();
28     }
29
30     1 referencia
31     private void btnRegistrarse_Click(object sender, EventArgs e)
32     {
33         Form3 formu = new Form3();
34         this.Hide();
35         formu.Show();
36     }
37 }
```

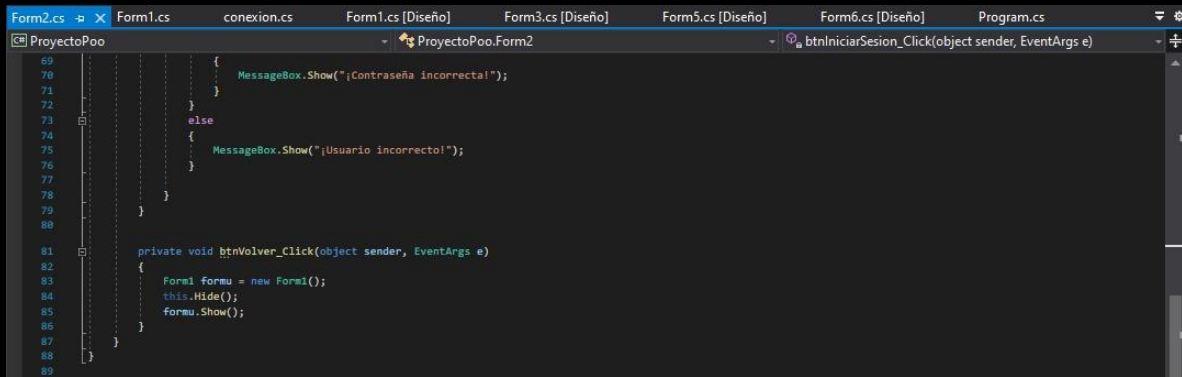
Aquí es donde muestra la primea pantalla, donde se puede registrar o iniciar sesion ya sea como cliente o proveedor.

Inicio de sesión.

```
11 namespace ProyectoPoo
12 {
13     8 referencias
14     public partial class Form2 : Form
15     {
16         3 referencias
17         public Form2()
18         {
19             InitializeComponent();
20         }
21
22         1 referencia
23         private void btnIniciarSesion_Click(object sender, EventArgs e)
24         {
25             string password;
26             string usuario = txtUsuario.Text;
27             string contra = txtContraseña.Text;
28             string url = "D:\\Documentos\\Poo\\Proyecto\\\" + usuario + ".txt";
```

```
Form2.cs x Form1.cs conexion.cs Form1.cs [Diseño] Form3.cs [Diseño] Form5.cs [Diseño] Form6.cs [Diseño] Program.cs
ProjectoPoo ProyectoPoo.Form2 btnIniciarSesion_Click(object sender, EventArgs e)
26 string url = "D:\\Documentos\\Poo\\Proyecto\\" + usuario + ".txt";
27
28 if (cbTipoCuenta.SelectedIndex == -1)
29 {
30     MessageBox.Show("Escoge un tipo de cuenta", "", MessageBoxButtons.OK);
31 }
32 else
33 {
34     if (cbTipoCuenta.SelectedIndex == 0)
35     {
36         url = "D:\\Documentos\\Poo\\Proyecto\\Cliente\\" + usuario + ".txt";
37     }
38     if (cbTipoCuenta.SelectedIndex == 1)
39     {
40         url = "D:\\Documentos\\Poo\\Proyecto\\Proveedor\\" + usuario + ".txt";
41     }
42
43     if (File.Exists(url))
44     {
45         password = File.ReadAllText(url);
46         if (contra.Equals(password))
47         {
48             MessageBox.Show("¡Ingreso exitoso, bienvenido!");
49
50             if (cbTipoCuenta.SelectedIndex == 0)
51             {
52                 //Se ingresa a la pantalla de clientes
```

```
Form2.cs x Form1.cs conexion.cs Form1.cs [Diseño] Form3.cs [Diseño] Form5.cs [Diseño] Form6.cs [Diseño] Program.cs
ProjectoPoo ProyectoPoo.Form2 btnIniciarSesion_Click(object sender, EventArgs e)
51 {
52     //Se ingresa a la pantalla de clientes
53
54     Form4 formu = new Form4();
55     this.Hide();
56     formu.Show();
57 }
58
59 if (cbTipoCuenta.SelectedIndex == 1)
60 {
61     //Se ingresa a la pantalla de proveedores
62
63     Form5 formu = new Form5();
64     this.Hide();
65     formu.Show();
66 }
67
68 else
69 {
70     MessageBox.Show("¡Contraseña incorrecta!");
71 }
72
73 else
74 {
75     MessageBox.Show("¡Usuario incorrecto!");
76 }
77
78 }
```



```
Form2.cs X Form1.cs conexion.cs Form1.cs [Diseño] Form3.cs [Diseño] Form5.cs [Diseño] Form6.cs [Diseño] Program.cs
ProyectoPoo ProyectoPoo.Form2 btnIniciarSesion_Click(object sender, EventArgs e)
69
70 {
71     MessageBox.Show("Contraseña incorrecta!");
72 }
73 else
74 {
75     MessageBox.Show("Usuario incorrecto!");
76 }
77 }
78 }
79
80
81 private void btnVolver_Click(object sender, EventArgs e)
82 {
83     Form1 formu = new Form1();
84     this.Hide();
85     formu.Show();
86 }
87
88
89 }
```

Aquí es donde el usuario elige como iniciar sesion si como cliente o como proveedor. Asi como el boton de volver y la comprobación de que la contraseña y el usuario ingresado es correcto. Y este a su vez lo envia a la siguiente pantalla.

Registrarse como nuevo usuario.

```
11 namespace ProyectoPoo
12 {
13     4 referencias
14     public partial class Form3 : Form
15     {
16         1 referencia
17         public Form3()
18         {
19             InitializeComponent();
20         }
21         1 referencia
22         private void btnRegistro_Click(object sender, EventArgs e)
23         {
24             string usuario = txtUsuario.Text;
25             string contra = txtContraseña.Text;
26             string url = "D:\\Documentos\\Poo\\Proyecto\\\" + usuario + ".txt";
27             if (cbTipoCuenta.SelectedIndex == -1)
28             {
29                 if (cbTipoCuenta.SelectedIndex == -1)
30                 {
31                     MessageBox.Show("Escoja un tipo de cuenta", "", MessageBoxButtons.OK);
32                 }
33             }
34             else
35             {
36                 if (cbTipoCuenta.SelectedIndex == 0)
37                 {
38                     url = "D:\\Documentos\\Poo\\Proyecto\\Cliente\\" + usuario + ".txt";
39                 }
40                 if (cbTipoCuenta.SelectedIndex == 1)
41                 {
42                     url = "D:\\Documentos\\Poo\\Proyecto\\Proveedor\\" + usuario + ".txt";
43                 }
44             }
45             if (File.Exists(url))
46             {
47                 MessageBox.Show("ERROR. ¡Usuario ya existe!");
48                 txtUsuario.Clear();
49                 txtContraseña.Clear();
50             }
51             else
52             {
53                 File.WriteAllText(url, contra);
54                 MessageBox.Show("Usuario Registrado con éxito");
55                 txtUsuario.Clear();
56                 txtContraseña.Clear();
57             }
58         }
59         1 referencia
60         private void btnVolver_Click(object sender, EventArgs e)
61         {
62             Form1 formu = new Form1();
63             this.Hide();
64             formu.Show();
65         }
66     }
67 }
```

Aquí es donde se crea una nueva cuenta, aquí puede seleccionar si registrarse

como proveedor o como cliente. También existe una comprobación para ver si el usuario ya existe.

Buscar producto.

```
11 namespace ProyectoPoo
12 {
13     4 referencias
14     public partial class Form4 : Form
15     {
16         private SqlConnection conn;
17         private string sCn;
18         private SqlDataAdapter dal;
19         private SqlDataReader dr1;
20
21         1 referencia
22         public Form4()
23         {
24             InitializeComponent();
25
26             conexion cn = new conexion();
27             cn.conec();
28             sCn = cn.cadena;
29             conn = new SqlConnection(sCn);
30
31         }
32
33         1 referencia
34         public void llenar()
35         {
36             DataSet ds = new DataSet();
37             SqlDataAdapter da = new SqlDataAdapter("SELECT Nombre FROM PRODUCTOS", conn);
38             da.Fill(ds, "PRODUCTOS");
39             cbProductos.DataSource = ds.Tables[0].DefaultView;
40             cbProductos.ValueMember = "Nombre";
41         }
42
43         1 referencia
44         private void label3_Click(object sender, EventArgs e)
45         {
46         }
47
48         1 referencia
49         private void Form4_Load(object sender, EventArgs e)
50         {
51             llenar();
52         }
53
54         1 referencia
55         private void cbProductos_SelectedIndexChanged(object sender, EventArgs e)
56         {
57             if (cbProductos.SelectedIndex != -1)
58             {
59                 try
60                 {
61                     conn.Open();
62                     string nom1 = cbProductos.Text;
63                     string seleccion = "SELECT * FROM PRODUCTOS WHERE Nombre = '" + nom1 + "'";
64                     dal = new SqlDataAdapter(seleccion, conn);
65                     SqlParameter prm = new SqlParameter("Nombre", SqlDbType.VarChar);
66                     prm.Value = nom1;
67                     dal.SelectCommand.Parameters.Add(prm);
68                     dr1 = dal.SelectCommand.ExecuteReader();
69                     while (dr1.Read())
70                     {
71                         nudCantidad.Maximum = Convert.ToInt32(dr1["cantidad"]);
72                         txtPrecio.Text = dr1["precio"].ToString();
73                     }
74                     conn.Close();
75                 }
76             }
77         }
78     }
79 }
```


Agregar productos.

```
12 namespace ProyectoPoo
13 {
14     4 referencias
15     public partial class Form5 : Form
16     {
17         private SqlConnection conn;
18         private SqlCommand insert1;
19         private string sCn;
20
21         1 referencia
22         public Form5()
23         {
24             InitializeComponent();
25
26             conexion cn = new conexion();
27
28             cn.conec();
29
30             sCn = cn.cadena;
31
32             conn = new SqlConnection(sCn);
33
34             conn.Open();
35
36         }
37
38         1 referencia
39         private void button1_Click(object sender, EventArgs e)
40         {
41             try
42             {
43                 string insertproducto;
44                 insertproducto = "INSERT INTO PRODUCTOS(Nombre, Descripcion, cantidad, precio)";
45                 insertproducto += "VALUES(@Nombre,@Descripcion,@cantidad,@precio)";
46                 insert1 = new SqlCommand(insertproducto, conn);
47                 insert1.Parameters.Add(new SqlParameter("@Nombre", SqlDbType.VarChar));
48                 insert1.Parameters["@Nombre"].Value = txtNombreP.Text;
49                 insert1.Parameters.Add(new SqlParameter("@Descripcion", SqlDbType.VarChar));
50                 insert1.Parameters["@Descripcion"].Value = txtDescripcionP.Text;
51                 insert1.Parameters.Add(new SqlParameter("@cantidad", SqlDbType.Int));
52                 insert1.Parameters["@cantidad"].Value = nUDCantidadP.Value;
53                 insert1.Parameters.Add(new SqlParameter("@precio", SqlDbType.Int));
54                 insert1.Parameters["@precio"].Value = txtPrecioP.Text;
55
56                 insert1.ExecuteNonQuery();
57
58                 txtNombreP.Text = "";
59                 txtDescripcionP.Text = "";
60                 nUDCantidadP.Value = 0;
61                 txtPrecioP.Text = "";
62
63                 MessageBox.Show("Producto agregado");
64                 conn.Close();
65             }
66             catch
67             {
68                 MessageBox.Show("Error");
69             }
70
71         }
72
73         1 referencia
74         private void button1_Click_1(object sender, EventArgs e)
75         {
76             Form2 formu = new Form2();
77             this.Hide();
78             formu.Show();
79         }
80     }
81 }
```

```
Form5.cs Form4.cs Form3.cs Form2.cs Form1.cs conexion.cs Form1.cs [Diseño] Form6.cs [Diseño] Program.cs
ProyectoPoo ProyectoPoo.Form5 button1_Click_1(object sender, EventArgs e)
35 private void button1_Click(object sender, EventArgs e)
36 {
37     try
38     {
39         string insertproducto;
40         insertproducto = "INSERT INTO PRODUCTOS(Nombre, Descripcion, cantidad, precio)";
41         insertproducto += "VALUES(@Nombre,@Descripcion,@cantidad,@precio)";
42         insert1 = new SqlCommand(insertproducto, conn);
43         insert1.Parameters.Add(new SqlParameter("@Nombre", SqlDbType.VarChar));
44         insert1.Parameters["@Nombre"].Value = txtNombreP.Text;
45         insert1.Parameters.Add(new SqlParameter("@Descripcion", SqlDbType.VarChar));
46         insert1.Parameters["@Descripcion"].Value = txtDescripcionP.Text;
47         insert1.Parameters.Add(new SqlParameter("@cantidad", SqlDbType.Int));
48         insert1.Parameters["@cantidad"].Value = nUDCantidadP.Value;
49         insert1.Parameters.Add(new SqlParameter("@precio", SqlDbType.Int));
50         insert1.Parameters["@precio"].Value = txtPrecioP.Text;
51
52         insert1.ExecuteNonQuery();
53
54         txtNombreP.Text = "";
55         txtDescripcionP.Text = "";
56         nUDCantidadP.Value = 0;
57         txtPrecioP.Text = "";
58
59         MessageBox.Show("Producto agregado");
60         conn.Close();
61     }
62     catch
63     {
64         MessageBox.Show("Error");
65     }
66 }
67
68 1 referencia
69 private void button1_Click_1(object sender, EventArgs e)
70 {
71     Form2 formu = new Form2();
72     this.Hide();
73     formu.Show();
74 }
```

```
Form5.cs Form4.cs Form3.cs Form2.cs Form1.cs conexion.cs Form1.cs [Diseño] Form6.cs [Diseño] Program.cs
ProyectoPoo ProyectoPoo.Form5 button1_Click_1(object sender, EventArgs e)
56 nUDCantidadP.Value = 0;
57 txtPrecioP.Text = "";
58
59 MessageBox.Show("Producto agregado");
60 conn.Close();
61 }
62 catch
63 {
64     MessageBox.Show("Error");
65 }
66 }
67
68 1 referencia
69 private void button1_Click_1(object sender, EventArgs e)
70 {
71     Form2 formu = new Form2();
72     this.Hide();
73     formu.Show();
74 }
75 }
76 }
```

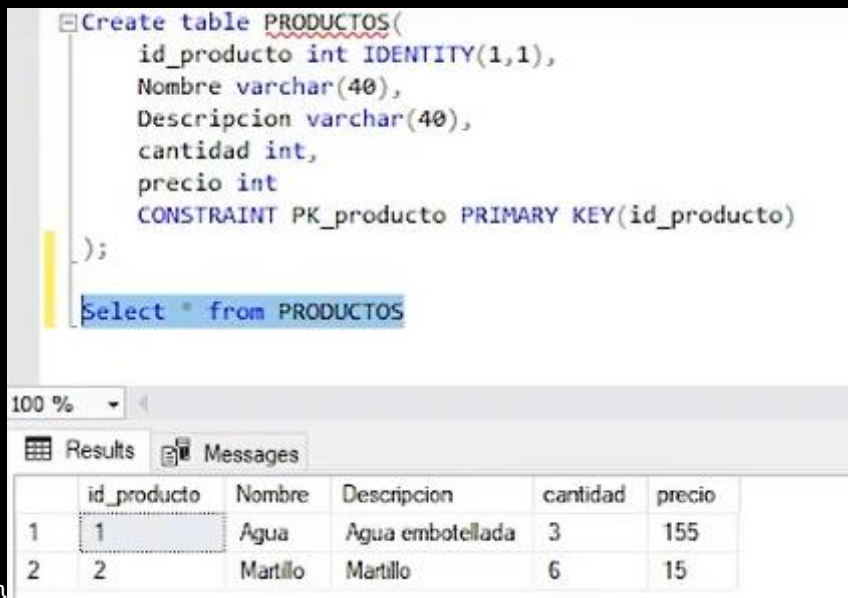
Aquí es donde el proveedor ingresa, el nombre, la cantidad y el precio de los productos.

Calcular el precio total.

```
10
11 namespace ProyectoPoo
12 {
13     4 referencias
14     public partial class Form6 : Form
15     {
16         1 referencia
17         public Form6(string nombre, string cantidad, string precio)
18         {
19             InitializeComponent();
20
21             lbProducto.Text += " " + nombre;
22             lbCantidad.Text += " " + cantidad;
23             lbPrecioIndividual.Text += " $" + precio;
24             lbPrecioTotal.Text += " $" + (Convert.ToInt32(precio) * Convert.ToInt32(cantidad));
25         }
26
27         1 referencia
28         private void Form6_Load(object sender, EventArgs e)
29         {
30         }
31     }
32 }
```

Aquí es donde se calcula el precio total que ha elegido el cliente.

Base de datos.



The screenshot shows a SQL query editor with two parts. The top part is a SQL script to create a table named 'PRODUCTOS' with columns: 'id_producto' (int, identity), 'Nombre' (varchar(40)), 'Descripcion' (varchar(40)), 'cantidad' (int), and 'precio' (int). A primary key constraint 'PK_producto' is defined on 'id_producto'. The bottom part shows the execution of a 'Select * from PRODUCTOS' query, resulting in a table with two rows of data.

```
Create table PRODUCTOS(  
    id_producto int IDENTITY(1,1),  
    Nombre varchar(40),  
    Descripcion varchar(40),  
    cantidad int,  
    precio int  
    CONSTRAINT PK_producto PRIMARY KEY(id_producto)  
);  
  
Select * from PRODUCTOS
```

	id_producto	Nombre	Descripcion	cantidad	precio
1	1	Agua	Agua embotellada	3	155
2	2	Martillo	Martillo	6	15

Aquí se almacena todos los datos
ingresados por el proveedor.

Desarrolladores:



Cruz Enrique Moreno Lozano.

Enzo Enrique Saravia Díaz.