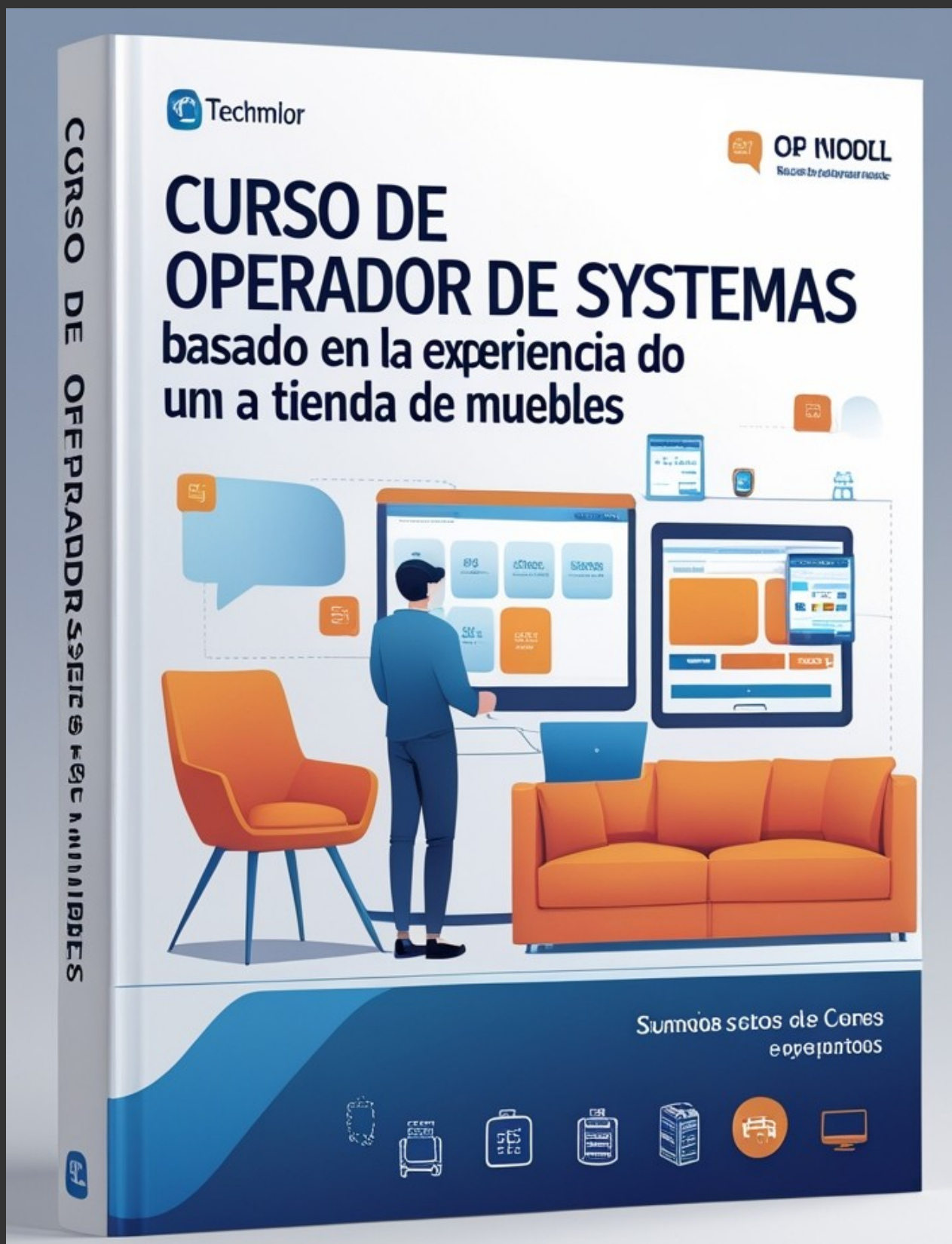


Guía de los Tres Formularios - Módulo 3: Configuración de Sistemas POS



¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

Guía de los Tres Formularios - Módulo 3: Configuración de Sistemas POS

Objetivo: Configurar y mantener un sistema POS para una tienda de muebles.

Lección 3.1: Instalación de hardware POS

- Instalación de escáneres y lectores de códigos de barras.
- Conexión y configuración de periféricos en el sistema.

Lección 3.2: Configuración del software POS

- Carga de bases de datos de productos y precios.
- Configuración de opciones de venta y facturación.

Ejercicio práctico 3.1: Instalación de un sistema POS básico

Este ejercicio te guiará paso a paso en la creación de un sistema POS básico utilizando **Visual Basic 2008**. Se incluyen tres formularios esenciales:

Formulario 1: Splash Screen

- Pantalla de carga inicial del sistema.
- Personalización con el logotipo y nombre del negocio.
- Tiempo de visualización antes de redirigir al login.

Formulario 2: Login

- Validación de usuario y contraseña.
- Conexión a una base de datos de usuarios.
- Seguridad básica con encriptación de contraseñas.

Formulario 3: MainForm (Interfaz Principal)

- Menú principal con opciones de ventas, inventario y configuración.
 - Accesos rápidos a funciones críticas del sistema POS.
 - Integración con la base de datos de productos y precios.
-

Simulador 3.1: Carga de bases de datos de productos y precios

- Ejercicio interactivo para cargar y gestionar una base de datos simulada.
 - Uso de formularios para agregar, modificar y eliminar productos.
 - Visualización de listados y búsquedas dentro del sistema POS.
-

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

Este material está basado en la experiencia real de un operador de sistemas en una tienda de muebles en 2005 y forma parte del libro "**Curso de Operador de Sistemas basado en una tienda de muebles**".

Puedes acceder a este contenido y seguir el desarrollo de cada formulario en www.hakatu.com, donde se irán publicando los avances paso a paso.

Resumen del flujo:

1. **SplashScreen**: Se muestra al inicio de la aplicación durante 3 segundos y luego oculta el formulario y muestra el **LoginForm**.
2. **LoginForm**: El usuario ingresa su nombre de usuario y contraseña. Si la combinación es correcta, se muestra el **MainForm**.
3. **MainForm**: Es el formulario principal donde podrás agregar la funcionalidad de tu aplicación.

Consideraciones:

- Asegúrate de reemplazar "Server=localhost\SQLEXPRESS;Database=TuBaseDeDatos;Integrated Security=True;" con la cadena de conexión correcta para tu base de datos.
- La tabla **usuarios** en la base de datos debe tener las columnas **NombreUsuario** y **Contraseña**.

1. SplashScreen

Este formulario se mostrará al iniciar la aplicación, luego de 3 segundos te llevará al **LoginForm**.

```
' SplashScreen.vb
Public Class SplashScreen
    Private Sub SplashScreen_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.FormBorderStyle = FormBorderStyle.None 'quitar bordes
        MainForm.Hide();
        Me.StartPosition = FormStartPosition.CenterScreen ' Centrar la pantalla
        Me.Size = New Size(800, 500) ' Ajustar tamaño (800x600 es solo un ejemplo)

        Timer1.Interval = 3000 ' 3 segundos
        Timer1.Start() ' Iniciar el temporizador
    End Sub

    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Timer1.Stop() ' Detener el temporizador
        Me.Hide() ' Ocultar el SplashScreen
        LoginForm.Show() ' Mostrar el formulario de login
    End Sub

    ' Evento para manejar el clic en el LinkLabel
    Private Sub LinkLabel1_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

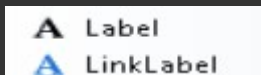
Aprende a resolver problemas comunes y mucho más.

```

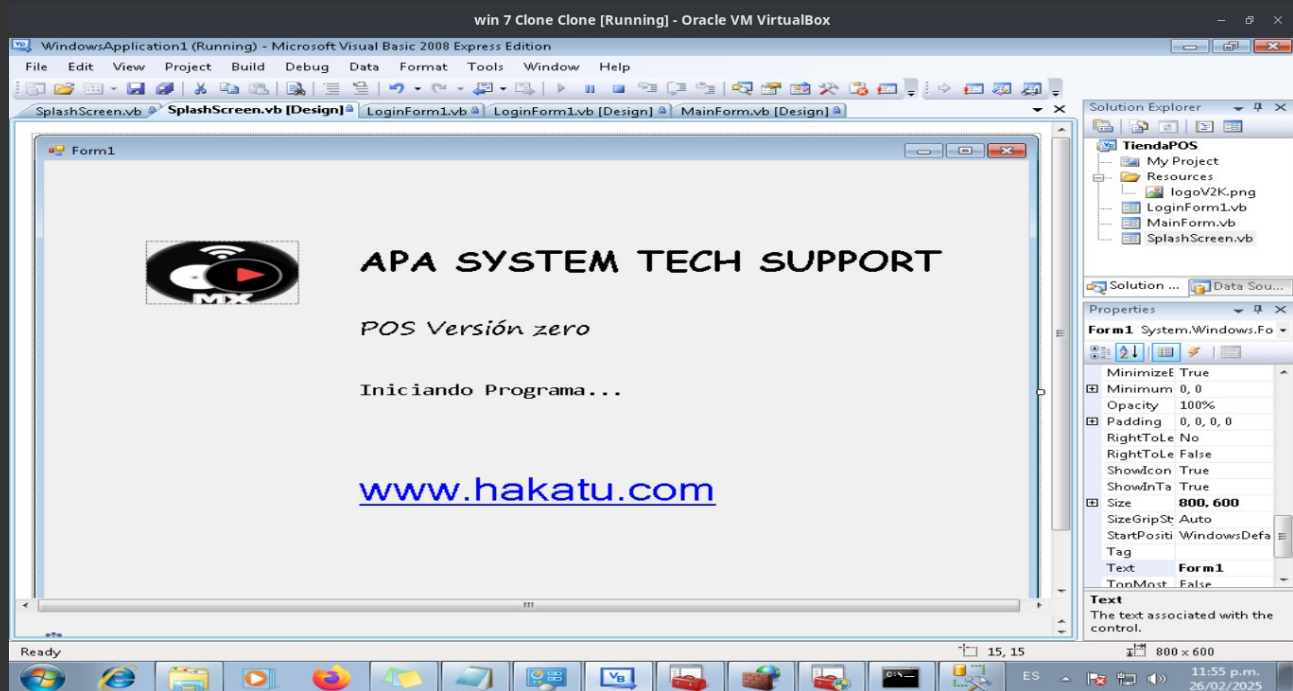
        Process.Start("http://www.hakatu.com")
    End Sub

End Class

```



propiedades: name, LinkLabelBehaviour, sizeMode: stretchImage



2. LoginForm

Este formulario permite al usuario ingresar un nombre de usuario y una contraseña. Verifica si la combinación es correcta y, si es así, te lleva al **MainForm**.

```

' LoginForm.vb
Public Class LoginForm

    Dim connectionString As String = "Server=localhost\
    SQLEXPRESS;Database=TuBaseDeDatos;Integrated Security=True;"

    ' Crear la conexión
    Dim connection As New SqlConnection(connectionString)

    Private Sub LoginButton_Click(sender As Object, e As EventArgs) Handles
    LoginButton.Click
        Dim query As String = "SELECT COUNT(*) FROM usuarios WHERE NombreUsuario
    = @Username AND Contraseña = @Password"
        ' Crear el comando SQL
        Dim command As New SqlCommand(query, connection)

```

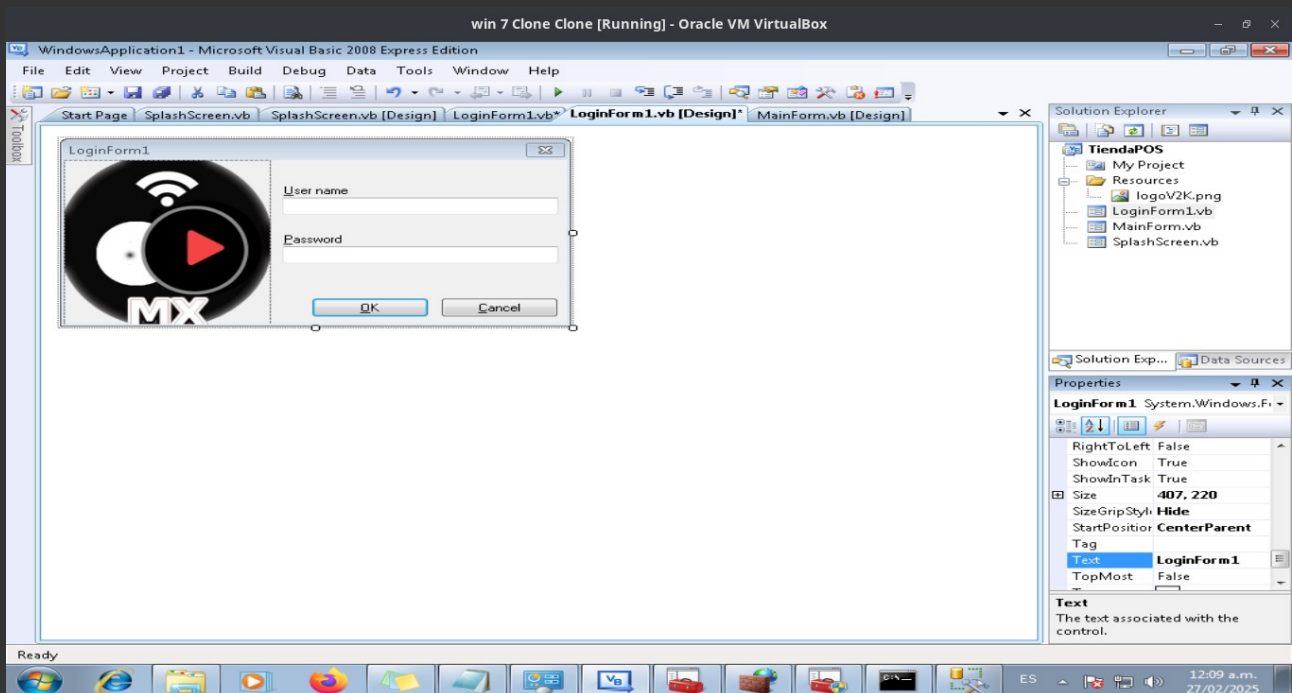
¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí](#)!
Aprende a resolver problemas comunes y mucho más.

```

' Agregar parámetros al comando SQL parecido a php mysql al ? Cada signo
es un parametro(@)
command.Parameters.AddWithValue("@Username", txtUsuario.Text)
command.Parameters.AddWithValue("@Password", txtContraseña.Text)

Try
    connection.Open() ' Abrir la conexión a la base de datos
    Dim result As Integer = Convert.ToInt32(command.ExecuteScalar()) '
Ejecutar el comando y obtener el resultado
    connection.Close()
    If result > 0 Then
        MessageBox.Show("Login successful!")
        Me.Hide() ' Ocultar el formulario de login
        MainForm.Show() ' Mostrar el formulario principal
    Else
        MessageBox.Show("Invalid username or password")
    End If
Catch ex As Exception
    MessageBox.Show("Error: " & ex.Message)
End Try
End Sub
End Class

```



3. MainForm

Este formulario se muestra cuando el usuario ha iniciado sesión correctamente. Aquí puedes colocar el contenido principal de la aplicación.

```

' MainForm.vb
Public Class MainForm
    Private Sub MainForm_Load(sender As Object, e As EventArgs) Handles
        MyBase.Load
        ' Puedes agregar aquí lo que necesitas cuando se carga el MainForm
    End Sub
End Class

```

¿Quieres aprender más sobre soporte técnico y otros temas?

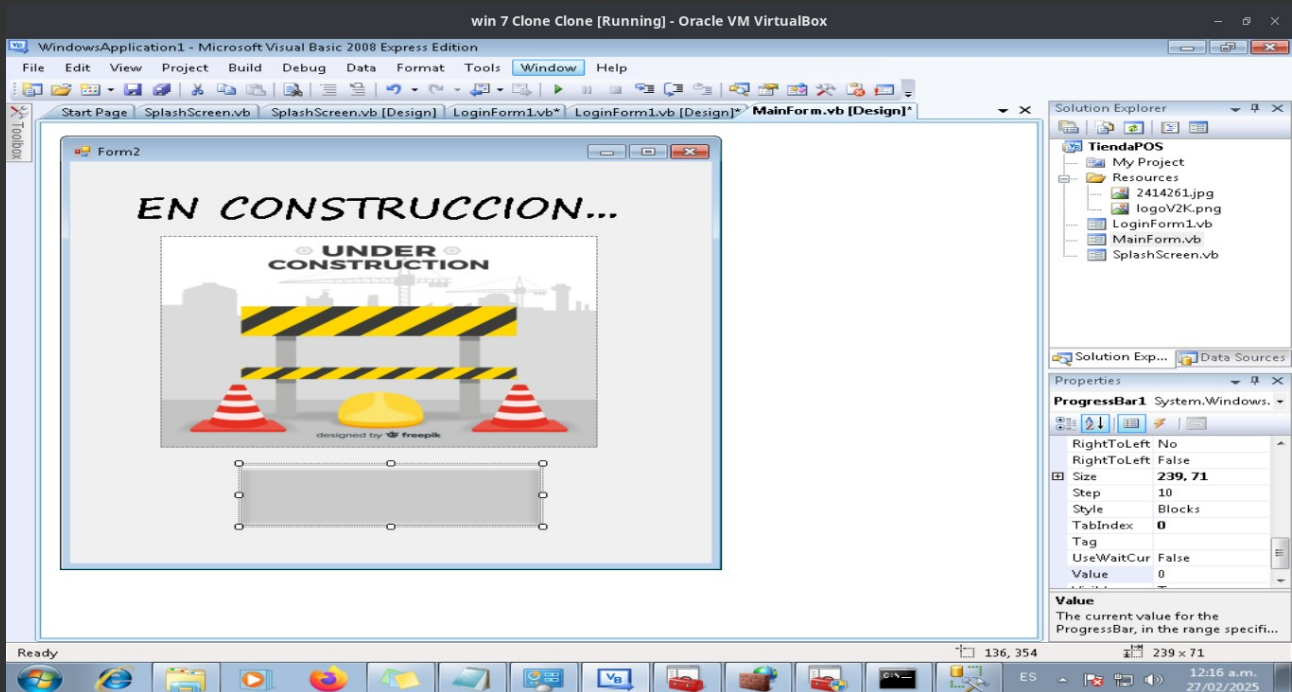
¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.


```

        Me.StartPosition = FormStartPosition.CenterScreen
        Me.Size = New Size(1024, 768) ' Ajustar el tamaño del formulario (puedes
cambiarlo según lo necesites)
    End Sub
End Class

```



Iniciar la aplicación sin un Main

Cuando inicias la aplicación, Visual Basic por defecto ejecuta el primer formulario. Dado que tienes los formularios **SplashScreen**, **LoginForm** y **MainForm**, debes configurar el **SplashScreen** como el formulario inicial, y luego navegar entre los demás formularios.

Para hacer esto, sigue estos pasos:

1. Abrir el formulario de inicio (SplashScreen):

- Ve a **Propiedades** del proyecto (haz clic derecho sobre el nombre del proyecto en el **Explorador de Soluciones** y selecciona **Propiedades**).
- En **Aplicación** busca la opción **Formulario de inicio** y selecciona **SplashScreen**.

Esto hará que al iniciar la aplicación, se cargue primero el **SplashScreen**. Luego de 3 segundos, se ocultará y se mostrará el **LoginForm**.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

MainForm (o "Interfaz Principal" del POS), y cómo integrar la búsqueda de productos en la base de datos, mostrar los resultados en un **DataGridView**, y calcular la suma total de los renglones del **DataGridView**.

Objetivo

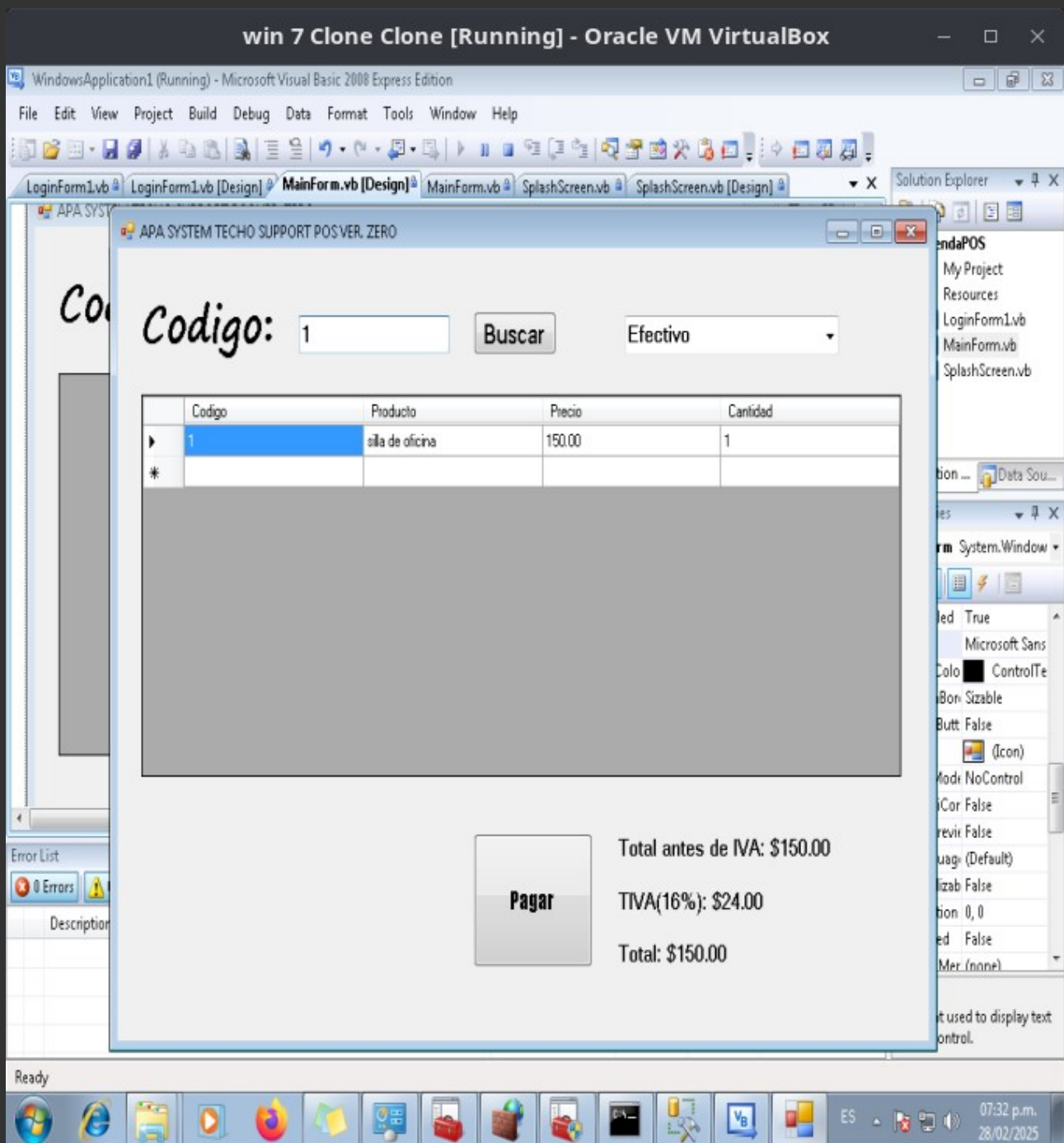
El propósito del formulario **MainForm** es proporcionar la interfaz donde los usuarios pueden interactuar con el sistema POS. Este formulario deberá permitir:

1. **Búsqueda de productos en la base de datos.**
2. **Mostrar los productos en un DataGridView.**
3. **Sumar el total de los renglones (productos)** para calcular el total de la compra.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.



Estructura del Formulario

Primero, en el formulario **MainForm**, vamos a necesitar:

- Un **TextBox** para que el usuario ingrese el código o nombre del producto que desea buscar.
- Un **Button** para iniciar la búsqueda.
- Un **DataGridView** para mostrar los resultados de la búsqueda (productos encontrados).
- Un **Label** o un **TextBox** para mostrar el total acumulado de los productos en el carrito.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

Paso 1: Crear la Interfaz

En Visual Basic, puedes usar el diseñador visual para agregar estos controles al formulario. Los elementos que necesitas son:

- **TextBox:** txtBuscarProducto (para ingresar el nombre o código del producto).
- **Button:** btnBuscarProducto (para ejecutar la búsqueda).
- **Button:** btnPagar (para pagar la compra).
- **DataGridView:** DataGridView1 (para mostrar los productos encontrados).
- **Label :** LabelTotal (para mostrar el total de la compra).
- **Label:** LabelSubTotal (para mostrar el total de la compra).
- **Label :** LabelIva (para mostrar el total de la compra).
- **Label :** lblCodigo (buscar producto).
- **Combo de opciones:** cmbTiposPago(efectivo,tarjeta,transferencia)

The screenshot shows a Windows form titled "Form1". At the top left, there is a label "Codigo:" in a stylized font. To its right is a text box for input. Further right is a button labeled "Buscar". To the right of the "Buscar" button is a dropdown menu labeled "Elegir tipo de pago". Below these elements is a large, empty rectangular area, likely intended for a DataGridView. At the bottom center, there is a button labeled "Pagar". To the right of the "Pagar" button, there are three vertically stacked "0" labels, which are likely placeholders for sub-total, total, and IVA labels.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

Paso 2: Código para Buscar el Producto y Llenar el DataGridView

Ahora, vamos a implementar la lógica que realiza la búsqueda del producto en la base de datos y muestra los resultados en el **DataGridView**.

1. Conectar a la base de datos

En primer lugar, debes tener una conexión a la base de datos. Usaremos **SqlConnection** y **SqlCommand** para hacer la consulta a la base de datos.

```
Imports System.Data.SqlClient

Public Class MainForm

    Dim connectionString As String = "Server=localhost\
SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"

    Dim connection As New SqlConnection(connectionString)

    Dim total As Decimal = 0
```

2. Usar SqlDataReader para Obtener los Productos

Luego, vamos a utilizar **SqlDataReader** para leer los datos de los productos y mostrar los resultados en el **DataGridView**. Para esto utilizamos el boton buscar o en el textbox buscar utilizamos el evento **TextChanged** para llamar al procedimiento **BuscarAgregarProducto(codigo)**.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Dim codigo As String = txtCodigo.Text

    BuscarAgregarProducto(codigo)

End Sub
```

```
Private Sub BuscarAgregarProducto(ByVal codigo As String)

    Dim query As String = "SELECT * FROM Productos WHERE ProductoID =@Codigo"

    Dim cmd As New SqlCommand(query, connection)

    cmd.Parameters.AddWithValue("@Codigo", codigo)

    Try
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```
connection.Open()
```

```
Dim reader As SqlDataReader = cmd.ExecuteReader()
```

```
If reader.HasRows Then
```

```
    reader.Read()
```

```
    ' si encuentra el producto lo agregamos al dtagrid
```

```
    'columnas de BD a variables
```

```
    Dim nombreProducto As String = reader("Nombre").ToString()
```

```
    Dim precioProducto As Decimal = Convert.ToDecimal(reader("Precio"))
```

```
    Dim cantidad As Integer = Convert.ToInt32(reader("Cantidad"))
```

```
    Dim codigoBarra As Integer = Convert.ToInt32(reader("ProductoID"))
```

```
    'esto se agrego por error en el for
```

```
    'erro al buscar producto:refrencia a objeto no establecida cmomo istancia d eobjeto
```

```
    'celdas no existen, estan vacias o debe agregarse antes al grid
```

```
    'validar que no haya columnas en cero antes
```

```
If DataGridView1.Columns.Count = 0 Then
```

```
    DataGridView1.Columns.Add("Codigo", "Codigo")
```

```
    DataGridView1.Columns.Add("Producto", "Producto")
```

```
    DataGridView1.Columns.Add("Precio", "Precio")
```

```
    DataGridView1.Columns.Add("Cantidad", "Cantidad")
```

```
End If
```

```
Dim encontrado As Boolean = False
```

```
For Each row As DataGridViewRow In DataGridView1.Rows
```

```
    ' si tiene algo algun valor hacer la conversion a string
```

```
    'nothing se refiera a que no se haya creado nnguna columna antes y por eso no hay nada
```

```
    'se refiera a que no haya estrucra creada antes en datagrid como fila sy columnas en  
este caso columnas
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        If row.Cells("Codigo").Value IsNot Nothing AndAlso
row.Cells("Codigo").Value.ToString() = codigo Then

            'si el producto ya existe solo aumentar la cantidad
            row.Cells("Cantidad").Value = Convert.ToInt32(row.Cells("Cantidad").Value) + 1

            encontrado = True

            Exit For
        End If
    Next

    If Not encontrado Then
        DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)
    End If

    'DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)

    ActualizarTotal()

Else
    MessageBox.Show("Producto no encontrado")

End If

Catch ex As Exception
    MessageBox.Show("Error al buscar el producto: " & ex.Message)
Finally
    connection.Close()

End Try

End Sub

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

text/plain;charset=utf-8

b5a

```
Private Sub BuscarAgregarProducto(ByVal codigo As String)
```

```
    Dim query As String = "SELECT * FROM Productos WHERE ProductoID =@Codigo"
```

```
    Dim cmd As New SqlCommand(query, connection)
```

```
    cmd.Parameters.AddWithValue("@Codigo", codigo)
```

```
    Try
```

```
        connection.Open()
```

```
        Dim reader As SqlDataReader = cmd.ExecuteReader()
```

```
        If reader.HasRows Then
```

```
            reader.Read()
```

```
            ' si encuentra el producto lo agregamos al dtagrid
```

```
            'columnas de BD a variables
```

```
            Dim nombreProducto As String = reader("Nombre").ToString()
```

```
            Dim precioProducto As Decimal = Convert.ToDecimal(reader("Precio"))
```

```
            Dim cantidad As Integer = Convert.ToInt32(reader("Cantidad"))
```

```
            Dim codigoBarra As Integer = Convert.ToInt32(reader("ProductoID"))
```

```
            'esto se agrego por error en el for
```

```
            'erro al buscar producto:refrencia a objeto no establecida cmomo istancia d eobjeto
```

```
            'celdas no existen, estan vacias o debe agregarse antes al grid
```

```
            'validar que no haya columnas en cero antes
```

```
            If DataGridView1.Columns.Count = 0 Then
```

```
                DataGridView1.Columns.Add("Codigo", "Codigo")
```

```
                DataGridView1.Columns.Add("Producto", "Producto")
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```
DataGridView1.Columns.Add("Precio", "Precio")
DataGridView1.Columns.Add("Cantidad", "Cantidad")
End If
```

```
Dim encontrado As Boolean = False
```

```
For Each row As DataGridViewRow In DataGridView1.Rows
```

```
    ' si tiene algo algun valor hacer la conversion a string
```

```
    'nothing se refiera a que no se haya creado nnguna columna antes y por eso no hay nada
```

```
    'se refiera a que no haya estrucra creada antes en datagrid como fila sy columanas en
este caso columnas
```

```
    If row.Cells("Codigo").Value IsNot Nothing AndAlso
row.Cells("Codigo").Value.ToString() = codigo Then
```

```
        'si el producto ya exste solo aumentar ka cantidad
```

```
        row.Cells("Cantidad").Value = Convert.ToInt32(row.Cells("Cantidad").Value) + 1
```

```
        encontrado = True
```

```
        Exit For
```

```
    End If
```

```
Next
```

```
If Not encontrado Then
```

```
    DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)
```

```
End If
```

```
'DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)
```

```
ActualizarTotal()
```

```
Else
```

```
    MessageBox.Show("Producto no encontrado")
```

```
End If
```

```
Catch ex As Exception
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.


```

        MessageBox.Show("Error al buscar el producto: " & ex.Message)
    Finally
        connection.Close()

    End Try

```

```
End Sub
```

3. Sumar los Renglones del DataGridView

Después de llenar el **DataGridView**, vamos a sumar los valores de los productos (por ejemplo, precio) para calcular el total de la compra. Cuando haya un cambio en cantidad se actualice el total con el procedimiento ActualizarTotal().

```

Private Sub DataGridView1_CellValueChanged(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellValueChanged
    If e.ColumnIndex = DataGridView1.Columns("Cantidad").Index Then
        ActualizarTotal()
    End If

End Sub

```

4. Llena el combo con los tipos de pago.

```

Private Sub CargarTiposPAgo()
    Try
        connection.Open()

        Dim query As String = "SELECT TiposPagoID,Descripcion FROM TiposPago"
        Dim cmd As New SqlCommand(query, connection)
        'para poder leer los resultados de la consulta
        Dim reader As SqlDataReader = cmd.ExecuteReader()

        'creamos un datatable para poder cargar los datos al combo elegir que columnas queremos que
        cargue
    
```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

Dim dt As New DataTable()
dt.Load(reader)
cmbTipooPago.DataSource = dt
cmbTipooPago.DisplayMember = "Descripcion"
cmbTipooPago.ValueMember = "TiposPagoID"
connection.Close()

Catch ex As Exception

    MessageBox.Show("Error al cargar los tipos de pago:" & ex.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

End Sub

```

4. boton pagar.

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnPagar.Click

```

```

    ActualizarTotal()

    Try

        connection.Open()

        'obtener el tipo del pago dle combo

        Dim tiposPagoID As Integer = cmbTipooPago.SelectedValue

        Dim query As String = "INSERT INTO ventas (Fecha,Total,TiposPagoID)OUTPUT
INSERTED.VentaID VALUES (@FechaVenta,@Total,@TiposPagoID)"

        Dim cmd As New SqlCommand(query, connection)

        cmd.Parameters.AddWithValue("@FechaVenta", DateTime.Now)
        cmd.Parameters.AddWithValue("@Total", total)

        If cmbTipooPago.SelectedIndex <> -1 Then

            cmd.Parameters.AddWithValue("@TiposPagoID", tiposPagoID)

            'cmd.ExecuteNonQuery()

        Else

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
    MessageBox.Show("Seleccione tipo de pago antes de continuar", "Advertencia",  
    MessageBoxButtons.OK, MessageBoxIcon.Warning)
```

```
    connection.Close()
```

```
End If
```

```
'Ejecutamos la consulta y obtenemos el ID de la venta scalar
```

```
'Dim queryVentaID As String = "SELECT SCOPE_IDENTITY()"
```

```
'Dim cmdVentaID As New SqlCommand(queryVentaID, connection)
```

```
Dim VentaID As Integer = Convert.ToInt32(cmd.ExecuteScalar())
```

```
lblCodigo.Text = VentaID.ToString()
```

```
If VentaID = 0 Then
```

```
    MessageBox.Show("Error: no se generó un ID de venta válido", "Error",  
    MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
End If
```

```
'insertar lo del datagrid
```

```
For Each row As DataGridViewRow In DataGridView1.Rows
```

```
    If Not row.IsNewRow Then 'evitamos la última fila vacía
```

```
        lblCodigo.Text = VentaID.ToString()
```

```
        Dim queryDetalle As String = "INSERT INTO DetalleVenta (VentaID,  
ProductoID,Cantidad,Precio) VALUES (@VentaID,@ProductoID,@Cantidad,@Precio)"
```

```
        Dim cmdDetalle As New SqlCommand(queryDetalle, connection)
```

```
'cmdDetalle.Parameters.AddWithValue("@VentaID", Convert.ToInt32(VentaID))
```

```
cmdDetalle.Parameters.AddWithValue("@VentaID", VentaID)
```

```
cmdDetalle.Parameters.AddWithValue("@ProductoID",  
Convert.ToInt32(row.Cells("Codigo").Value)) ' es la columna del datagridview que contiene  
ProductoID
```

```
cmdDetalle.Parameters.AddWithValue("@Cantidad",  
Convert.ToInt32(row.Cells("Cantidad").Value))
```

```
cmdDetalle.Parameters.AddWithValue("@Precio",  
Convert.ToDecimal(row.Cells("Precio").Value))
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        cmdDetalle.ExecuteNonQuery()

        'Restar la cantidad del producto vendido

        Dim queryActualizarStock As String = "UPDATE Productos SET Cantidad = Cantidad
- @Cantidad WHERE ProductoID = @ProductoID"

        Dim cmdActualizarStock As New SqlCommand(queryActualizarStock, connection)

        cmdActualizarStock.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

        cmdActualizarStock.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value))

        cmdActualizarStock.ExecuteNonQuery()

    End If

Next

connection.Close()

MessageBox.Show("Venta Guarda OK.", "Exito", MessageBoxButtons.OK,
MessageBoxIcon.Information)

Catch ex As Exception

    MessageBox.Show("Error al guardar la venta: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)

Finally

    If connection.State = ConnectionState.Open Then

        connection.Close()

    End If

End Try

End Sub

End Class

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

Controles y Componentes:

1. MainForm (Formulario Principal)

- Evento: Load (MainForm_Load)
- Propiedad usada: AutoSizeColumnsMode (de DataGridView1)

2. Button1 (Botón para agregar producto)

- Evento: Click (Button1_Click)
- Evento: KeyDown (Button1_KeyDown)

3. txtCodigo (Caja de texto para ingresar código del producto)

- Evento: TextChanged (txtCodigo_TextChanged)

4. DataGridView1 (Tabla de productos agregados)

- Evento: CellValueChanged (DataGridView1_CellValueChanged)
- Propiedades usadas:
 - Columns (para agregar columnas si no existen)
 - Rows (para iterar sobre las filas)
 - AutoSizeColumnsMode

5. LabelSubTotal (Etiqueta para subtotal)

- Propiedad usada: Text (para mostrar el subtotal)

6. LabelIva (Etiqueta para IVA)

- Propiedad usada: Text (para mostrar el IVA)

7. LabelTotal (Etiqueta para total)

- Propiedad usada: Text (para mostrar el total con IVA)

8. cmbTipoPago (ComboBox para tipo de pago)

- Propiedades usadas:
 - Items.Add() (para agregar opciones)
 - SelectedIndex (para seleccionar el primer ítem)
 - DataSource (para cargar tipos de pago desde la base de datos)
 - DisplayMember
 - ValueMember

9. BtnPagar (Botón para registrar la venta)

- Evento: Click (Button2_Click o BtnPagar_Click)

10. lblCodigo (Etiqueta para mostrar el ID de la venta generada)

- Propiedad usada: Text

11. SqlConnection connection (Conexión a la base de datos)

- Métodos usados:
 - Open()

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

- Close()

12. **SqlCommand cmd** (Comandos SQL)

- **Métodos usados:**
 - ExecuteReader()
 - ExecuteScalar()
 - ExecuteNonQuery()

13. **SqlDataReader reader** (Lector de datos SQL)

- **Método usado:** Read()

Eventos Utilizados:

1. Button1_Click → Ejecuta BuscarAgregarProducto(txtCodigo.Text)
2. Button1_KeyDown → Permite agregar producto con Enter
3. txtCodigo_TextChanged → Busca el producto automáticamente al escribir
4. MainForm_Load → Configura el DataGridView y carga tipos de pago
5. DataGridView1_CellValueChanged → Actualiza total cuando cambia la cantidad de productos
6. BtnPagar_Click → Registra la venta en la base de datos

Código Ejemplo para MainForm.vb

```
Imports System.Data.SqlClient
```

```
Public Class MainForm
```

```
    Dim connectionString As String = "Server=localhost\
SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"
```

```
    Dim connection As New SqlConnection(connectionString)
```

```
    Dim total As Decimal = 0
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
```

```
        Dim codigo As String = txtCodigo.Text
```

```
        BuscarAgregarProducto(codigo)
```

```
    End Sub
```

```
    Private Sub BuscarAgregarProducto(ByVal codigo As String)
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        Dim query As String = "SELECT * FROM Productos WHERE
ProductoID =@Codigo"

        Dim cmd As New SqlCommand(query, connection)
        cmd.Parameters.AddWithValue("@Codigo", codigo)

    Try
        connection.Open()
        Dim reader As SqlDataReader = cmd.ExecuteReader()
        If reader.HasRows Then
            reader.Read()

            ' si encuentra el producto lo agregamos al dtagrid
            'columnas de BD a variables
            Dim nombreProducto As String =
reader("Nombre").ToString()
            Dim precioProducto As Decimal =
Convert.ToDecimal(reader("Precio"))
            Dim cantidad As Integer =
Convert.ToInt32(reader("Cantidad"))
            Dim codigoBarra As Integer =
Convert.ToInt32(reader("ProductoID"))

            'esto se agrego por error en el for
            'erro al buscar producto:referencia a objeto no
establecida cmomo instancia d eobjeto
            'celdas no existren, estan vacias o debe agregarse
antes al grid
            'validar que no haya columnas en cero antes
            If DataGridView1.Columns.Count = 0 Then

                DataGridView1.Columns.Add("Codigo", "Codigo")

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        DataGridView1.Columns.Add("Producto",
"Producto")

        DataGridView1.Columns.Add("Precio", "Precio")
        DataGridView1.Columns.Add("Cantidad",
"Cantidad")

    End If

    Dim encontrado As Boolean = False
    For Each row As DataGridViewRow In
DataGridView1.Rows
        ' si tiene algo algun valor hacer la
conversion a string
        'nothing se refiera a que no se haya creado
ninguna columna antes y por eso no hay nada
        'se refiera a que no haya estrucra creada
antes en datagrid como fila sy columnas en este caso columnas
        If row.Cells("Codigo").Value IsNot Nothing
AndAlso row.Cells("Codigo").Value.ToString() = codigo Then
            'si el producto ya exste solo aumentar ka
cantidad

            row.Cells("Cantidad").Value =
Convert.ToInt32(row.Cells("Cantidad").Value) + 1
            encontrado = True
            Exit For
        End If
    Next

    If Not encontrado Then
        DataGridView1.Rows.Add(codigoBarra,
nombreProducto, precioProducto, 1)
    End If

    'DataGridView1.Rows.Add(codigoBarra,
nombreProducto, precioProducto, 1)

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.


```

        ActualizarTotal()
    Else
        MessageBox.Show("Producto no encontrado")

    End If

    Catch ex As Exception
        MessageBox.Show("Error al buscar el pdrocuto: " &
ex.Message)
    Finally
        connection.Close()

    End Try

End Sub

Private Sub ActualizarTotal()
    Dim subtotal As Decimal = 0

    Dim tasaIVA As Decimal = 0.16
    Dim Iva As Decimal = 0

    For Each row As DataGridViewRow In DataGridView1.Rows
        If row.Cells("Precio").Value IsNot Nothing AndAlso
row.Cells("Cantidad").Value IsNot Nothing Then
            subtotal +=
Convert.ToDecimal(row.Cells("Precio").Value) *
Convert.ToInt32(row.Cells("Cantidad").Value)
        End If
    Next

    Iva = subtotal * tasaIVA

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
        total = subtotal + Iva
        LabelSubTotal.Text = "Total antes de IVA: " &
subtotal.ToString("C")
        LabelIva.Text = "TIVA(16%): " & Iva.ToString("C")
        LabelTotal.Text = "Total: " & total.ToString("C")
```

End Sub

```
Private Sub MainForm_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
```

```
    DataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'expandir la scolumas por
igual en todo el datagrid
```

```
    'agregarlos al combo
```

```
    cmbTipooPago.Items.Add("Efectivo")
```

```
    cmbTipooPago.Items.Add("Tarjeta")
```

```
    cmbTipooPago.Items.Add("Transferencia")
```

```
    cmbTipooPago.SelectedIndex = 0
```

```
    CargarTiposPAgo()
```

End Sub

```
Private Sub txtCodigo_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtCodigo.TextChanged
```

```
    Dim codigo As String = txtCodigo.Text
```

```
    BuscarAgregarProducto(codigo)
```

End Sub

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

Private Sub Button1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Button1.KeyDown
    If e.KeyCode = Keys.Enter Then
        Dim codigo As String = txtCodigo.Text
        BuscarAgregarProducto(codigo)
        e.SuppressKeyPress = True 'no beep
    End If
End Sub

```

```

Private Sub DataGridView1_CellValueChanged(ByVal sender As
Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridView1.CellValueChanged
    If e.ColumnIndex = DataGridView1.Columns("Cantidad").Index
Then
        ActualizarTotal()
    End If
End Sub

```

```

Private Sub CargarTiposPAgo()
    Try
        connection.Open()
        Dim query As String = "SELECT TiposPagoID,Descripcion
FROM TiposPago"
        Dim cmd As New SqlCommand(query, connection)
        'para poder leer los resulatdos de la consulta
        Dim reader As SqlDataReader = cmd.ExecuteReader()
        'creamos un datable para poder cargar los dato sal
combo elegir que columnas quereonso que cargue
        Dim dt As New DataTable()
        dt.Load(reader)
    End Try
End Sub

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        cmbTipooPago.DataSource = dt
        cmbTipooPago.DisplayMember = "Descripcion"
        cmbTipooPago.ValueMember = "TiposPagoID"
        connection.Close()

    Catch ex As Exception
        MessageBox.Show("Error al cargar los tipos de pago:" &
ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

    End Try

End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnPagar.Click
    ActualizarTotal()
    Try
        connection.Open()
        'obtener el tipo del pago dle combo
        Dim tiposPagoID As Integer =
cmbTipooPago.SelectedValue

        Dim query As String = "INSERT INTO ventas
(Fecha,Total,TiposPagoID)OUTPUT INSERTED.VentaID VALUES
(@FechaVenta,@Total,@TiposPagoID)"

        Dim cmd As New SqlCommand(query, connection)
        cmd.Parameters.AddWithValue("@FechaVenta",
DateTime.Now)
        cmd.Parameters.AddWithValue("@Total", total)

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        If cmbTipooPago.SelectedIndex <> -1 Then
            cmd.Parameters.AddWithValue("@TiposPagoID",
tiposPagoID)

            'cmd.ExecuteNonQuery()
        Else
            MessageBox.Show("Seleccione tipo de pago antes de
continuar", "Advertencia", MessageBoxButtons.OK,
MessageBoxIcon.Warning)

            connection.Close()
        End If

        'Ekecutamos la consulata y obtneims elID de la venta
scalar
        'Dim queryVentaID As String = "SELECT
SCOPE_IDENTITY()"
        'Dim cmdVentaID As New SqlCommand(queryVentaID,
connection)

        Dim VentaID As Integer =
Convert.ToInt32(cmd.ExecuteNonQuery())

        lblCodigo.Text = VentaID.ToString()

        If VentaID = 0 Then
            MessageBox.Show("Error: no se genero UN ID de
venta vakido", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End If

        'insertar lo del datagrid
        For Each row As DataGridViewRow In DataGridView1.Rows
            If Not row.IsNewRow Then 'evitamos la utima fila
vacía

                lblCodigo.Text = VentaID.ToString()
            End If
        End For
    End Sub
End Class

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        Dim queryDetalle As String = "INSERT INTO
DetalleVenta (VentaID, ProductoID,Cantidad,Precio) VALUES
(@VentaID,@ProductoID,@Cantidad,@Precio)"

        Dim cmdDetalle As New SqlCommand(queryDetalle,
connection)

'cmdDetalle.Parameters.AddWithValue("@VentaID",
Convert.ToInt32(VentaID))

        cmdDetalle.Parameters.AddWithValue("@VentaID",
VentaID)

cmdDetalle.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value)) ' es la columna del
datagridview que contien ProdcutoID

cmdDetalle.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

        cmdDetalle.Parameters.AddWithValue("@Precio",
Convert.ToDecimal(row.Cells("Precio").Value))

        cmdDetalle.ExecuteNonQuery()

'Restar la cantidad del producto vendido

        Dim queryActualizarStock As String = "UPDATE
Productos SET Cantidad = Cantidad - @Cantidad WHERE ProductoID =
@ProductoID"

        Dim cmdActualizarStock As New
SqlCommand(queryActualizarStock, connection)

cmdActualizarStock.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

cmdActualizarStock.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value))

        cmdActualizarStock.ExecuteNonQuery()

End If

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        Next
        connection.Close()
        MessageBox.Show("Venta Guarda OK.", "Exito",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    Catch ex As Exception
        MessageBox.Show("Error al guardar la venta: " &
        ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Finally
        If connection.State = ConnectionState.Open Then
            connection.Close()
        End If
    End Try
End Sub
End Class

```

```

text/plain; charset=utf-8

```

```

2740

```

```

Imports System.Data.SqlClient

```

```

Public Class MainForm

```

```

    Dim connectionString As String = "Server=localhost\
SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"

```

```

    Dim connection As New SqlConnection(connectionString)

```

```

    Dim total As Decimal = 0

```

```

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click

```

```

        Dim codigo As String = txtCodigo.Text

```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        BuscarAgregarProducto(codigo)
    End Sub

    Private Sub BuscarAgregarProducto(ByVal codigo As String)

        Dim query As String = "SELECT * FROM Productos WHERE
ProductoID =@Codigo"

        Dim cmd As New SqlCommand(query, connection)
        cmd.Parameters.AddWithValue("@Codigo", codigo)

        Try
            connection.Open()

            Dim reader As SqlDataReader = cmd.ExecuteReader()

            If reader.HasRows Then
                reader.Read()

                ' si encuentra el producto lo agregamos al dtagrid
                'columnas de BD a variables
                Dim nombreProducto As String =
reader("Nombre").ToString())
                Dim precioProducto As Decimal =
Convert.ToDecimal(reader("Precio"))
                Dim cantidad As Integer =
Convert.ToInt32(reader("Cantidad"))
                Dim codigoBarra As Integer =
Convert.ToInt32(reader("ProductoID"))

                'esto se agrego por error en el for
                'erro al buscar producto:refrencia a objeto no
establecida cmomo istancia d eobjeto
                'celdas no existen, estan vacias o debe agregarse
antes al grid
                'validar que no haya columnas en cero antes

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.


```

        If DataGridView1.Columns.Count = 0 Then

            DataGridView1.Columns.Add("Codigo", "Codigo")
            DataGridView1.Columns.Add("Producto",
"Producto")
            DataGridView1.Columns.Add("Precio", "Precio")
            DataGridView1.Columns.Add("Cantidad",
"Cantidad")

        End If

        Dim encontrado As Boolean = False
        For Each row As DataGridViewRow In
DataGridView1.Rows
            ' si tiene algo algun valor hacer la
conversion a string
            'nothing se refiera a que no se haya creado
ninguna columna antes y por eso no hay nada
            'se refiera a que no haya estrucra creada
antes en datagrid como fila sy columnas en este caso columnas
            If row.Cells("Codigo").Value IsNot Nothing
AndAlso row.Cells("Codigo").Value.ToString() = codigo Then
                'si el producto ya exste solo aumentar ka
cantidad

                row.Cells("Cantidad").Value =
Convert.ToInt32(row.Cells("Cantidad").Value) + 1
                encontrado = True
                Exit For
            End If
        Next

        If Not encontrado Then
            DataGridView1.Rows.Add(codigoBarra,
nombreProducto, precioProducto, 1)

```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        End If

        'DataGridView1.Rows.Add(codigoBarra,
nombreProducto, precioProducto, 1)

        ActualizarTotal()
    Else
        MessageBox.Show("Producto no encontrado")

    End If

    Catch ex As Exception
        MessageBox.Show("Error al buscar el pdrocuto: " &
ex.Message)
    Finally
        connection.Close()

    End Try

End Sub

Private Sub ActualizarTotal()
    Dim subtotal As Decimal = 0

    Dim tasaIVA As Decimal = 0.16
    Dim Iva As Decimal = 0

    For Each row As DataGridViewRow In DataGridView1.Rows
        If row.Cells("Precio").Value IsNot Nothing AndAlso
row.Cells("Cantidad").Value IsNot Nothing Then
            subtotal +=
Convert.ToDecimal(row.Cells("Precio").Value) *
Convert.ToInt32(row.Cells("Cantidad").Value)

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        End If
    Next

    Iva = subtotal * tasaIVA
    total = subtotal + Iva

    LabelSubTotal.Text = "Total antes de IVA: " &
subtotal.ToString("C")

    LabelIva.Text = "TIVA(16%): " & Iva.ToString("C")
    LabelTotal.Text = "Total: " & total.ToString("C")

End Sub

Private Sub MainForm_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

    DataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'expandir la scolumnas por
igual en todo el datagrid

    'agregarlos al combo
    cmbTipooPago.Items.Add("Efectivo")
    cmbTipooPago.Items.Add("Tarjeta")
    cmbTipooPago.Items.Add("Transferencia")
    cmbTipooPago.SelectedIndex = 0

    CargarTiposPAgo()

End Sub

Private Sub txtCodigo_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtCodigo.TextChanged

    Dim codigo As String = txtCodigo.Text

    BuscarAgregarProducto(codigo)

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

End Sub

```
Private Sub Button1_KeyDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles Button1.KeyDown
```

```
    If e.KeyCode = Keys.Enter Then
```

```
        Dim codigo As String = txtCodigo.Text
```

```
        BuscarAgregarProducto(codigo)
```

```
        e.SuppressKeyPress = True 'no beep
```

```
    End If
```

End Sub

```
Private Sub DataGridView1_CellValueChanged(ByVal sender As  
Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)  
Handles DataGridView1.CellValueChanged
```

```
    If e.ColumnIndex = DataGridView1.Columns("Cantidad").Index  
Then
```

```
        ActualizarTotal()
```

```
    End If
```

End Sub

```
Private Sub CargarTiposPAgo()
```

```
    Try
```

```
        connection.Open()
```

```
        Dim query As String = "SELECT TiposPagoID, Descripcion  
FROM TiposPago"
```

```
        Dim cmd As New SqlCommand(query, connection)
```

```
        'para poder leer los resultados de la consulta
```

```
        Dim reader As SqlDataReader = cmd.ExecuteReader()
```

```
        'creamos un datable para poder cargar los datos al  
combo elegir que columnas queremos que cargue
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        Dim dt As New DataTable()
        dt.Load(reader)
        cmbTipooPago.DataSource = dt
        cmbTipooPago.DisplayMember = "Descripcion"
        cmbTipooPago.ValueMember = "TiposPagoID"
        connection.Close()

    Catch ex As Exception
        MessageBox.Show("Error al cargar los tipos de pago:" &
ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

    End Try

End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnPagar.Click
    ActualizarTotal()
    Try
        connection.Open()
        'obtener el tipo del pago dle combo
        Dim tiposPagoID As Integer =
cmbTipooPago.SelectedValue

        Dim query As String = "INSERT INTO ventas
(Fecha,Total,TiposPagoID)OUTPUT INSERTED.VentaID VALUES
(@FechaVenta,@Total,@TiposPagoID)"

        Dim cmd As New SqlCommand(query, connection)

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        cmd.Parameters.AddWithValue("@FechaVenta",
DateTime.Now)
        cmd.Parameters.AddWithValue("@Total", total)
        If cmbTipooPago.SelectedIndex <> -1 Then
            cmd.Parameters.AddWithValue("@TiposPagoID",
tiposPagoID)
            'cmd.ExecuteNonQuery()
        Else
            MessageBox.Show("Seleccione tipo de pago antes de
continuar", "Advertencia", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            connection.Close()
        End If

        'Ejecutamos la consulata y obtneims elID de la venta
scalar
        'Dim queryVentaID As String = "SELECT
SCOPE_IDENTITY()"
        'Dim cmdVentaID As New SqlCommand(queryVentaID,
connection)

        Dim VentaID As Integer =
Convert.ToInt32(cmd.ExecuteScalar())
        lblCodigo.Text = VentaID.ToString()
        If VentaID = 0 Then
            MessageBox.Show("Error: no se genero UN ID de
venta vakido", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End If

        'insertar lo del datagrid
        For Each row As DataGridViewRow In DataGridView1.Rows
            If Not row.IsNewRow Then 'evitamos la utima fila
vacía

```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

        lblCodigo.Text = VentaID.ToString()

        Dim queryDetalle As String = "INSERT INTO
DetalleVenta (VentaID, ProductoID,Cantidad,Precio) VALUES
(@VentaID,@ProductoID,@Cantidad,@Precio)"

        Dim cmdDetalle As New SqlCommand(queryDetalle,
connection)

'cmdDetalle.Parameters.AddWithValue("@VentaID",
Convert.ToInt32(VentaID))

        cmdDetalle.Parameters.AddWithValue("@VentaID",
VentaID)

cmdDetalle.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value)) ' es la columna del
datagridview que contien ProdcutoID

cmdDetalle.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

        cmdDetalle.Parameters.AddWithValue("@Precio",
Convert.ToDecimal(row.Cells("Precio").Value))

        cmdDetalle.ExecuteNonQuery()

'Restar la cantidad del producto vendido

        Dim queryActualizarStock As String = "UPDATE
Productos SET Cantidad = Cantidad - @Cantidad WHERE ProductoID =
@ProductoID"

        Dim cmdActualizarStock As New
SqlCommand(queryActualizarStock, connection)

cmdActualizarStock.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

cmdActualizarStock.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value))

        cmdActualizarStock.ExecuteNonQuery()

End If

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

        Next
        connection.Close()

        MessageBox.Show("Venta Guarda OK.", "Exito",
        MessageBoxButtons.OK, MessageBoxIcon.Information)

        Catch ex As Exception

            MessageBox.Show("Error al guardar la venta: " &
            ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

        Finally

            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If

        End Try

    End Sub

End Class

```

Descripción del Código

1. Conexión a la Base de Datos:

- La cadena de conexión se establece en `connectionString`, apuntando a tu base de datos.
- Usamos `SqlConnection` para abrir la conexión.

2. Búsqueda del Producto:

- En el evento `btnBuscarProducto_Click`, se toma el texto ingresado en el **TextBox** `txtBuscarProducto` y se pasa a la función `BuscarProducto`.
- La consulta SQL utiliza `LIKE` para buscar productos que contengan la palabra clave (nombre o ID del producto).

3. `SqlDataReader`:

- Se ejecuta la consulta y se usa `SqlDataReader` para leer los resultados.
- Se agregan los resultados al **DataGridView** `dgvProductos`, mostrando las columnas `ProductoID`, `NombreProducto`, y `Precio`.

4. Sumar el Total:

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

- Después de llenar el **DataGridView**, la función `CalcularTotal` recorre todas las filas y suma los valores de la columna `Precio`.
- El total se muestra en el **Label** `lblTotal`.

Paso 3: Configurar el DataGridView

Asegúrate de que el **DataGridView** tenga las siguientes columnas configuradas:

- `ProductoID` (Codigo)
- `NombreProducto`
- `Precio`
- `Cantidad`

Estas columnas se llenarán con los datos de la base de datos cuando realices la búsqueda.

Ejemplo de Configuración del DataGridView:

1. Abre el formulario en el diseñador de Visual Basic.
2. Selecciona el **DataGridView** `dgvProductos` y en la ventana de propiedades, ve a la propiedad **Columns**.
3. Añade las siguientes columnas:
 - **ProductoID** (tipo: `DataGridViewTextBoxColumn`).
 - **NombreProducto** (tipo: `DataGridViewTextBoxColumn`).
 - **Precio** (tipo: `DataGridViewTextBoxColumn`).

Resumen del Flujo:

1. El usuario ingresa un término de búsqueda en el **TextBox** `txtBuscarProducto`.
2. Al hacer clic en el botón **btnBuscarProducto**, se ejecuta la búsqueda en la base de datos.
3. Los resultados de la búsqueda se muestran en el **DataGridView**.
4. El total de la compra se calcula sumando los precios de los productos en el **DataGridView** y se muestra en el **Label** `lblTotal`.

Con esto, tienes una interfaz básica para tu sistema POS, con capacidad de buscar productos y calcular el total de la compra. ¡Continúa con el desarrollo y agrega más funcionalidades como agregar productos al carrito, eliminar, y generar facturas!

EL CODIGO COMPLETO

```
Imports System.Data.SqlClient
```

```
Public Class MainForm
```

```
    Dim connectionString As String = "Server=localhost\
    SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

Dim connection As New SqlConnection(connectionString)
Dim total As Decimal = 0

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Dim codigo As String = txtCodigo.Text
    BuscarAgregarProducto(codigo)
End Sub

```

```

Private Sub BuscarAgregarProducto(ByVal codigo As String)

    Dim query As String = "SELECT * FROM Productos WHERE ProductoID =@Codigo"
    Dim cmd As New SqlCommand(query, connection)
    cmd.Parameters.AddWithValue("@Codigo", codigo)

```

```

Try
    connection.Open()
    Dim reader As SqlDataReader = cmd.ExecuteReader()
    If reader.HasRows Then
        reader.Read()

        ' si encuentra el producto lo agregamos al dtagrid
        'columnas de BD a variables
        Dim nombreProducto As String = reader("Nombre").ToString()
        Dim precioProducto As Decimal = Convert.ToDecimal(reader("Precio"))
        Dim cantidad As Integer = Convert.ToInt32(reader("Cantidad"))
        Dim codigoBarra As Integer = Convert.ToInt32(reader("ProductoID"))

        'esto se agrego por error en el for
        'erro al buscar producto:refrencia a objeto no establecida cmomo istancia d eobjeto
        'celdas no existen, estan vacias o debe agregarse antes al grid
        'validar que no haya columnas en cero antes

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
If DataGridView1.Columns.Count = 0 Then
```

```
    DataGridView1.Columns.Add("Codigo", "Codigo")
```

```
    DataGridView1.Columns.Add("Producto", "Producto")
```

```
    DataGridView1.Columns.Add("Precio", "Precio")
```

```
    DataGridView1.Columns.Add("Cantidad", "Cantidad")
```

```
End If
```

```
Dim encontrado As Boolean = False
```

```
For Each row As DataGridViewRow In DataGridView1.Rows
```

```
    ' si tiene algo algun valor hacer la conversion a string
```

```
    'nothing se refiera a que no se haya creado ninguna columna antes y por eso no hay nada
```

```
    'se refiera a que no haya estructura creada antes en datagrid como fila y columnas en este caso columnas
```

```
    If row.Cells("Codigo").Value IsNot Nothing AndAlso  
row.Cells("Codigo").Value.ToString() = codigo Then
```

```
        'si el producto ya existe solo aumentar la cantidad
```

```
        row.Cells("Cantidad").Value = Convert.ToInt32(row.Cells("Cantidad").Value) + 1
```

```
        encontrado = True
```

```
    Exit For
```

```
End If
```

```
Next
```

```
If Not encontrado Then
```

```
    DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)
```

```
End If
```

```
'DataGridView1.Rows.Add(codigoBarra, nombreProducto, precioProducto, 1)
```

```
ActualizarTotal()
```

```
Else
```

```
    MessageBox.Show("Producto no encontrado")
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```
End If
Catch ex As Exception
    MessageBox.Show("Error al buscar el pdrocut: " & ex.Message)
Finally
    connection.Close()
End Try
End Sub
```

```
Private Sub ActualizarTotal()
    Dim subtotal As Decimal = 0
    Dim tasaIVA As Decimal = 0.16
    Dim Iva As Decimal = 0

    For Each row As DataGridViewRow In DataGridView1.Rows
        If row.Cells("Precio").Value IsNot Nothing AndAlso row.Cells("Cantidad").Value IsNot
Nothing Then
            subtotal += Convert.ToDecimal(row.Cells("Precio").Value) *
Convert.ToInt32(row.Cells("Cantidad").Value)
        End If
    Next

    Iva = subtotal * tasaIVA
    total = subtotal + Iva

    LabelSubTotal.Text = "Total antes de IVA: " & subtotal.ToString("C")
    LabelIva.Text = "TIVA(16%): " & Iva.ToString("C")
    LabelTotal.Text = "Total: " & total.ToString("C")

End Sub
```

¿Quieres aprender más sobre soporte técnico y otros temas?
¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
Private Sub MainForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    DataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill  
'expandir la scolumas por igual en todo el datagrid
```

```
    'agregarlos al combo
```

```
    cmbTipooPago.Items.Add("Efectivo")
```

```
    cmbTipooPago.Items.Add("Tarjeta")
```

```
    cmbTipooPago.Items.Add("Transferencia")
```

```
    cmbTipooPago.SelectedIndex = 0
```

```
    CargarTiposPAgo()
```

```
End Sub
```

```
Private Sub txtCodigo_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtCodigo.TextChanged
```

```
    Dim codigo As String = txtCodigo.Text
```

```
    BuscarAgregarProducto(codigo)
```

```
End Sub
```

```
Private Sub Button1_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Button1.KeyDown
```

```
    If e.KeyCode = Keys.Enter Then
```

```
        Dim codigo As String = txtCodigo.Text
```

```
        BuscarAgregarProducto(codigo)
```

```
        e.SuppressKeyPress = True 'no beep
```

```
    End If
```

```
End Sub
```

```
Private Sub DataGridView1_CellValueChanged(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellValueChanged
```

```
    If e.ColumnIndex = DataGridView1.Columns("Cantidad").Index Then
```

```
        ActualizarTotal()
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

End If

End Sub

Private Sub CargarTiposPAgo()

Try

connection.Open()

Dim query As String = "SELECT TiposPagoID,Descripcion FROM TiposPago"

Dim cmd As New SqlCommand(query, connection)

'para poder leer los resulatdos de la consulta

Dim reader As SqlDataReader = cmd.ExecuteReader()

'creamos un databale para poder cargar los dato sal combo elegir que columnas queremos que cargue

Dim dt As New DataTable()

dt.Load(reader)

cmbTipooPago.DataSource = dt

cmbTipooPago.DisplayMember = "Descripcion"

cmbTipooPago.ValueMember = "TiposPagoID"

connection.Close()

Catch ex As Exception

MessageBox.Show("Error al cargar los tipos de pago:" & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnPagar.Click

ActualizarTotal()

Try

connection.Open()

'obtener el tipo del pago dle combo

¿Quieres aprender más sobre soporte técnico y otros temas?
¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```

Dim tiposPagoID As Integer = cmbTipooPago.SelectedValue

Dim query As String = "INSERT INTO ventas (Fecha,Total,TiposPagoID)OUTPUT
INSERTED.VentaID VALUES (@FechaVenta,@Total,@TiposPagoID)"

Dim cmd As New SqlCommand(query, connection)

cmd.Parameters.AddWithValue("@FechaVenta", DateTime.Now)

cmd.Parameters.AddWithValue("@Total", total)

If cmbTipooPago.SelectedIndex <> -1 Then

    cmd.Parameters.AddWithValue("@TiposPagoID", tiposPagoID)

    'cmd.ExecuteNonQuery()

Else

    MessageBox.Show("Seleccione tipo de pago antes de continuar", "Advertencia",
    MessageBoxButtons.OK, MessageBoxIcon.Warning)

    connection.Close()

End If

'Ejecutamos la consulata y obtnems elID de la venta scalar

'Dim queryVentaID As String = "SELECT SCOPE_IDENTITY()"

'Dim cmdVentaID As New SqlCommand(queryVentaID, connection)

Dim VentaID As Integer = Convert.ToInt32(cmd.ExecuteNonQuery())

lblCodigo.Text = VentaID.ToString()

If VentaID = 0 Then

    MessageBox.Show("Error: no se genero UN ID de venta vakido", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

'insertar lo del datagrid

For Each row As DataGridViewRow In DataGridView1.Rows

    If Not row.IsNewRow Then 'evitamos la utima fila vacia

        lblCodigo.Text = VentaID.ToString()

        Dim queryDetalle As String = "INSERT INTO DetalleVenta (VentaID,
        ProductoID,Cantidad,Precio) VALUES (@VentaID,@ProductoID,@Cantidad,@Precio)"

```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```

Dim cmdDetalle As New SqlCommand(queryDetalle, connection)

'cmdDetalle.Parameters.AddWithValue("@VentaID", Convert.ToInt32(VentaID))

cmdDetalle.Parameters.AddWithValue("@VentaID", VentaID)

cmdDetalle.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value)) ' es la columna del atagridview que contien
ProdcutoID

cmdDetalle.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

cmdDetalle.Parameters.AddWithValue("@Precio",
Convert.ToDecimal(row.Cells("Precio").Value))

cmdDetalle.ExecuteNonQuery()

'Restar la cantidad del producto vendido

Dim queryActualizarStock As String = "UPDATE Productos SET Cantidad = Cantidad
- @Cantidad WHERE ProductoID = @ProductoID"

Dim cmdActualizarStock As New SqlCommand(queryActualizarStock, connection)

cmdActualizarStock.Parameters.AddWithValue("@Cantidad",
Convert.ToInt32(row.Cells("Cantidad").Value))

cmdActualizarStock.Parameters.AddWithValue("@ProductoID",
Convert.ToInt32(row.Cells("Codigo").Value))

cmdActualizarStock.ExecuteNonQuery()

End If

Next

connection.Close()

MessageBox.Show("Venta Guarda OK.", "Exito", MessageBoxButtons.OK,
MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show("Error al guardar la venta: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)

Finally

```

¿Quieres aprender más sobre soporte técnico y otros temas?
 ¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.


```
        If connection.State = ConnectionState.Open Then
            connection.Close()
        End If
    End Try
End Sub
```

LA TABLAS DE LA BD

use TiendaPOS;

```
create table Productos (
    ProductoID INT Primary key identity,
    Nombre nvarchar(100),
    Precio DECIMAL(10,2),
    Cantidad INT
);
```

```
CREATE TABLE Ventas(
    VentaID int primary key identity,
    Fecha DATETIME,
    Total DECIMAL(10,2),
    TipoPagoID varchar(50),
    FOREIGN KEY (TipoPagoID) REFERENCES TiposPago(TipoPagoID);
);
```

¿Quieres aprender más sobre soporte técnico y otros temas?
¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
CREATE TABLE DetalleVenta(  
DetalleVentaID int primary key identity,  
VentaID int,  
ProductoID int,  
Cantidad int,  
Precio DECIMAL(10,2),  
FOREIGN KEY (VentaID) REFERENCES Ventas(VentaID),  
FOREIGN KEY (ProductoID) REFERENCES Productos(ProductoID)  
);
```

```
exec sp_help DetalleVenta
```

```
create table Usuarios (  
UsuarioID int primary key identity,  
NombreUsuario nvarchar(50) unique,  
Contrasena nvarchar(100),  
Rol nvarchar(20)  
  
);
```

```
select top 1 * from Ventas order by VentaID desc  
select * from DetalleVenta where VentaID=25  
select * from Productos where ProductoID=1
```

```
EXEC sp_rename 'ventas.tipopafod', 'TiposPagoID', 'COLUMN';
```

¿Quieres aprender más sobre soporte técnico y otros temas?
¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.

```
insert into usuarios (NombreUsuario, Contraseña, Rol)
values ('admin','1234','Administrador');
```

```
insert into Productos(nombre, Precio, Cantidad)
values('silla de oficina',150,10)
```

```
add TipoPagoID varchar(50);
```

```
create table TiposPago(
TipoPagoID int primary key identity(1,1),
Descripcion varchar(50)
);
```

```
alter table Ventas
```

```
add constraint FK_TipoPago FOREIGN KEY (TipoPagoID) REFERENCES
TiposPago(TipoPagoID);
```

```
EXEC sp_rename 'ventas.tipospagos', 'TipoPagoID', 'COLUMN';
```

Recapitulación Completa para Resolver el Error de Conexión a SQL Server:

Resumen:

1. Agregar la referencia a **System.Data** en tu proyecto.
2. Importar la librería **System.Data.SqlClient**.
3. Verifica que el servicio de SQL Server esté en ejecución.
4. Habilita las conexiones remotas en SQL Server.
5. Asegúrate de que el puerto 1433 esté abierto en el firewall.
6. Habilita **TCP/IP** y **Named Pipes** en SQL Server.
7. Asegúrate de que el modo de autenticación esté configurado correctamente.
8. Verifica la conectividad con Telnet.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

9. Desactiva temporalmente el antivirus si es necesario.
10. Verifica la cadena de conexión en tu aplicación.
11. Probar la conexión desde tu aplicación con código.
12. Verifica la conexión con `sqlcmd` en CMD para asegurarte de que la conexión esté funcionando antes de probar la aplicación.

Nota importante: Si realizas cambios en la configuración (como habilitar TCP/IP, cambiar el modo de autenticación, o habilitar conexiones remotas), **es recomendable reiniciar el servicio de SQL Server Express** para que los cambios surtan efecto. Puedes reiniciar el servicio desde el **SQL Server Configuration Manager** o el **Administrador de servicios de Windows**.

Errores Comunes y Recomendaciones

1. Coincidencia de tipos de datos:

- Asegúrate de que los valores que insertas sean del tipo de dato correcto según la tabla. Por ejemplo, si una columna es `DECIMAL(10, 2)`, no puedes insertar texto.

2. Coincidencia de nombres de columnas en el DataGridView:

- Los nombres de las columnas en el `DataGridView` deben coincidir exactamente con los de la tabla en la base de datos para evitar errores y confusiones.

Recomendaciones:

- Verifica los tipos de datos al insertar registros.
- Usa nombres de columnas consistentes entre el `DataGridView` y la base de datos.

1. Agregar la referencia a System.Data en tu proyecto:

- Si aún no lo has hecho, es necesario agregar la referencia a **System.Data** para poder usar `SqlConnection` y otras funcionalidades de ADO.NET:
 - En Visual Studio, haz clic derecho en **Referencias** dentro del **Explorador de soluciones**.
 - Selecciona **Agregar referencia**.
 - Busca **System.Data** en la lista de ensamblados y marca la casilla.
 - Haz clic en **Aceptar**.

2. Importar la librería System.Data.SqlClient:

- Después de agregar la referencia a **System.Data**, importa la librería **SqlClient** al principio de tu archivo de código para poder usar las clases necesarias para conectar con SQL Server:

```
Imports System.Data.SqlClient
```

3. Verificar que el servicio de SQL Server esté en ejecución:

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

- Asegúrate de que el servicio de SQL Server esté activo. Puedes verificarlo desde el **SQL Server Configuration Manager** o desde el **Administrador de servicios** en Windows (services.msc).
- **Si el servicio no está en ejecución**, reinícialo siguiendo estos pasos:
 - Abre **SQL Server Configuration Manager**.
 - Ve a **SQL Server Services**.
 - Si el servicio **SQL Server (SQLEXPRESS)** está detenido, haz clic derecho sobre él y selecciona **Iniciar**.

4. Habilitar las conexiones remotas en SQL Server:

- Abre **SQL Server Management Studio (SSMS)**, conéctate al servidor.
- Ve a **Propiedades del servidor > Conexiones** y asegúrate de que la opción **Permitir conexiones remotas a este servidor** esté activada.
- **Reiniciar el servicio de SQL Server Express** (si has realizado cambios en la configuración):
 - Abre **SQL Server Configuration Manager** y ve a **SQL Server Services**.
 - Haz clic derecho sobre **SQL Server (SQLEXPRESS)** y selecciona **Reiniciar**.

5. Asegurarse de que el puerto 1433 esté abierto en el firewall:

- Verifica en el **Firewall de Windows** que el puerto TCP 1433 esté abierto para las conexiones de SQL Server:
 - Abre **Panel de Control > Sistema y seguridad > Firewall de Windows > Configuración avanzada**.
 - En **Reglas de entrada**, busca si existe una regla para permitir el tráfico en el puerto 1433. Si no existe, crea una nueva.

6. Habilitar TCP/IP y Named Pipes en SQL Server:

- Abre el **SQL Server Configuration Manager**.
- En **SQL Server Network Configuration > Protocols for SQLEXPRESS**, asegúrate de que tanto **TCP/IP** como **Named Pipes** estén habilitados.
- **Si realizas cambios, reinicia el servicio de SQL Server Express** para que los cambios surtan efecto:
 - Ve a **SQL Server Services**, haz clic derecho sobre **SQL Server (SQLEXPRESS)** y selecciona **Reiniciar**.

7. Comprobar que el modo de autenticación esté configurado correctamente (SQL Server y Windows):

- En **SSMS**, haz clic derecho sobre el servidor y selecciona **Propiedades**.
- En **Seguridad**, asegúrate de que el modo de autenticación esté configurado en **SQL Server y autenticación de Windows**.
- **Reinicia el servicio de SQL Server Express** si has cambiado el modo de autenticación:
 - Abre **SQL Server Configuration Manager** y ve a **SQL Server Services**.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

- Haz clic derecho sobre **SQL Server (SQLEXPRESS)** y selecciona **Reiniciar**.

8. Verificar la conectividad con Telnet:

- Abre **CMD** y usa el comando `telnet` para verificar si puedes conectarte al puerto 1433 de tu servidor:

```
telnet 127.0.0.1 1433
```

- Si no puedes conectarte, verifica el firewall o configura la red de tu servidor.

9. Desactivar el antivirus si es necesario:

- Si tienes un antivirus activo, puede estar bloqueando la conexión. Intenta desactivarlo temporalmente para ver si eso soluciona el problema.

10. Verificar la cadena de conexión en tu aplicación:

- En tu aplicación, asegúrate de que la cadena de conexión sea correcta y coincida con lo que aparece en **SSMS**:

```
Dim connectionString As String = "Server=localhost\
SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"
```

11. Probar la conexión desde tu aplicación con código:

- Utiliza el siguiente código en tu aplicación para probar la conexión:

```
Imports System.Data.SqlClient

Public Sub TestConnection()
    Dim connectionString As String = "Server=localhost\
SQLEXPRESS;Database=TiendaPOS;Integrated Security=True;"
    Dim connection As New SqlConnection(connectionString)

    Try
        connection.Open()
        MessageBox.Show("Conexión exitosa!")
    Catch ex As Exception
        MessageBox.Show("Error de conexión: " & ex.Message)
    Finally
        connection.Close()
    End Try
End Sub
```

12. Verificar la conexión con sqlcmd en CMD:

- Abre la ventana de **CMD** (Símbolo del sistema):
 - Presiona **Win + R**, escribe `cmd` y presiona Enter.
- Ejecuta el siguiente comando en CMD para verificar la conexión:


```
sqlcmd -S localhost\SQLEXPRESS -e
```
- `-S localhost\SQLEXPRESS`: Especifica el nombre del servidor y la instancia de SQL Server.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

- -e: Activa la salida de errores para obtener información detallada sobre la conexión.
- Si la conexión es exitosa, verás un prompt de SQL donde podrás escribir consultas.
- Si la conexión falla, revisa el mensaje de error para identificar el problema.

Aquí tienes un breve resumen de los errores que has encontrado al desarrollar tu sistema POS, junto con sus soluciones o causas:

Errores y soluciones en el desarrollo del POS

1. DataGridView sin encabezados

- **Error:** No se mostraban las columnas en `DataGridView1`.
- **Causa:** No se habían agregado manualmente.
- **Solución:** Se añadieron con `DataGridView1.Columns.Add(...)`.

2. Error al obtener **SCOPE_IDENTITY()**

- **Error:** *"No se puede convertir un objeto DBNULL en otros tipos"*.
- **Causa:** `ExecuteScalar()` devolvía `DBNull`.
- **Solución:** Se validó que `cmdVentaID.ExecuteScalar()` no fuera `DBNull` antes de convertirlo a `Int32`.

3. Error en la consulta **detalleventa**

- **Error:** Se pensaba que no insertaba datos, pero sí lo hacía.
- **Causa:** Posible error en la consulta o en la validación de datos.
- **Solución:** Se verificó en la base de datos y se confirmó que los datos se insertaban correctamente.

4. Carga del **cmbTipoDePago** desde la base de datos

- **Motivo de uso de DataTable:**
 - Permite manejar los datos en memoria antes de cargarlos en el `ComboBox`.
 - Facilita la vinculación (`DataSource`, `DisplayMember`, `ValueMember`).
- **Solución:** Se usó un `DataTable` para poblar el `ComboBox` con los valores correctos desde la base de datos.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.