

Aparcamiento  
MovimientoCuentasCorriente  
MovimientoCuentasCorriente.v2  
Libreria\_v2  
Gestion Articulos

\*\*\*\*\*

Declaracion de array

Dim numeros(19) As Integer  
Dim arrayBidimensional(filas, columnas) As TipoDeDato

.....

\_Load

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles Me.Load

Salto de Linea

Dim mensaje As String = "Hola," & vbCrLf & "este es un salto de línea."

.....

Focus()

KeyDown

Private Sub TextBox1\_KeyDown(sender As Object, e As KeyEventArgs) Handles  
TextBox1.KeyDown  
If e.KeyCode = Keys.Enter Then  
InputText  
Dim inputText As String  
inputText = InputBox("Introduce un valor:", "Input Box")

.....

For

For i = 0 To ListBox1.Items.Count - 1  
If ListBox1.SelectedIndex <> -1 Then  
btnAbajo.Enabled = True  
btnDerecha.Enabled = True  
btnIzq.Enabled = False  
End If  
Next

ComboBox

lbName.Items.Add(cbName.Text)  
cbName.Text = ""

Select Case

Dim pais As String = ListBox1.SelectedItem.ToString()  
  
Select Case pais  
Case "Alemania"  
txtPais.Text = "Alemania"  
.....  
Case "España"  
txtPais.Text = "España"  
.....  
Case "Italia"

```

        txtPais.Text = "Italia"
        .....
    Case "Inglaterra"
        txtPais.Text = "Inglaterra"
        .....
End Select

Select Case cantidad
    Case 1
        'Asignamos el precioUnitario al precioCliente ya que no hay descuento
        precioCliente = precioUnitario
        txtDTO.Text = "0%"
    Case 2, 3
        'Realizamos el calculo segun el porcentaje 3%
        precioCliente = (precioUnitario * cantidad) - ((precioUnitario * cantidad) * 0.03)
        txtDTO.Text = "3%"
    Case 4, 5, 6, 7
        'Realizamos el calculo segun el porcentaje 7%
        precioCliente = (precioUnitario * cantidad) - ((precioUnitario * cantidad) * 0.07)
        txtDTO.Text = "7%"
    Case Else
        'Realizamos el calculo segun el porcentaje 15%
        precioCliente = (precioUnitario * cantidad) - ((precioUnitario * cantidad) * 0.15)
        txtDTO.Text = "15%"
End Select

```

---

PictureBox

---

ListBox

```

    If txtNombreAlumno.Text.Equals("") Then
        MessageBox.Show("No puede estar vacio el nombre del alumno")
    Else
        lbAlumno.Items.Add(txtNombreAlumno.Text)
        lbOptativas.Items.Add(cbAsigOpt.SelectedItem)
    End If

    If lbAlumno.SelectedIndex <> -1 Then
        lbOptativas.Items.RemoveAt(lbAlumno.SelectedIndex)
        lbAlumno.Items.RemoveAt(lbAlumno.SelectedIndex)
    ElseIf lbOptativas.SelectedIndex <> -1 Then
        lbAlumno.Items.RemoveAt(lbOptativas.SelectedIndex)
        lbOptativas.Items.Remove(lbOptativas.SelectedItem)
    End If

    lbAlumno.Items.Clear()

```

---

Scroll

```

    PictureBox1.Size = New Size(169, VScrollBar1.Value)
    PictureBox1.Size = New Size(HScrollBar1.Value, 108)
    - Tener en cuenta el minimo y maximo del scroll

```

- Tener en cuenta vertical y horizontal

Image

```
PictureBox4.Image = Image.FromFile(Application.StartupPath & "\fotos_animales\
foto1.jpg")
```

---

Module Module1

```
Public variable As Date = Now
```

```
Public valor As String
```

```
Sub fecha()
```

```
    valor = Format(variable, "d/m/yy").ToString
```

```
    Form1.txtFecha.Visible = True
```

```
    Form1.txtHora.Visible = False
```

```
    Form1.txtFecha.Text = valor
```

```
End Sub
```

```
Sub hora()
```

```
    valor = Format(variable, "h:m:s")
```

```
    Form1.txtFecha.Visible = False
```

```
    Form1.txtHora.Visible = True
```

```
    Form1.txtHora.Text = valor
```

```
End Sub
```

```
End Module
```

```
Public Class Form1
```

```
    Private Sub btnFecha_Click(sender As Object, e As EventArgs) Handles btnFecha.Click
```

```
        Call fecha()
```

```
    End Sub
```

```
    Private Sub btnHora_Click(sender As Object, e As EventArgs) Handles btnHora.Click
```

```
        Call hora()
```

```
    End Sub
```

```
    Private Sub txtHora_TextChanged(sender As Object, e As EventArgs) Handles
```

```
txtHora.TextChanged
```

```
    End Sub
```

```
End Class
```

---

Module Module1

```
Public formulario1 As New Form1()
```

```
Public formulario2 As New Form2()
```

```
End Module
```

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

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

```
        Me.Hide()
```

```
        formulario2.TextBox1.Text = "pepepepepepe"
```

```
        formulario2.Button1.BackColor = Color.Red
```

```

        formulario2.Show()
    End Sub
End Class
Public Class Form2
    Inherits System.Windows.Forms.Form
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Hide()
        formulario1.Button1.BackColor = Color.Purple
        formulario1.Show()
    End Sub
End Class

```

---

```

Private Function ValidarNumero(message As String, messageTitle As String)
    Dim cantidad As Decimal
    Dim input As String

    Do
        input = InputBox(message, messageTitle)

        If Not Decimal.TryParse(input, cantidad) OrElse cantidad < 0 Then
            MessageBox.Show("Por favor introduce un número valido y positivo")
        End If
    
```

```

    Loop Until Decimal.TryParse(input, cantidad) AndAlso cantidad >= 0
    Return cantidad
End Function

```

```

Private Function ValidarTexto(message As String, messageTitle As String)
    Dim nombre As String

    Do
        nombre = InputBox(message, messageTitle)
        If String.IsNullOrEmpty(nombre) Then
            MessageBox.Show("El nombre no puede estar vacio, Error de entrada")
        ElseIf Not Regex.IsMatch(nombre, "^[a-zA-Z\s]+$") Then
            MessageBox.Show("El nombre solo puede hacer contener letra y espacios")
        End If
        Loop Until Not String.IsNullOrEmpty(nombre) AndAlso Regex.IsMatch(nombre, "^[a-zA-Z\s]+$")
    
```