

# CODIGO CLIENTE

Module Module1

```
Sub Main()
    Dim Oempleado As New Empleado
    Dim sn = "S"
    Dim opimp As Integer
    Do
        Try
            Console.Write("Ingrese Nombre: ")
            Oempleado.nombre = Console.ReadLine()
            Console.Write("Ingrese Cantidad de Hijos: ")
            Oempleado.Numero_Hijos = CInt(Console.ReadLine())
            Console.Write("Ingrese DNI: ")
            Oempleado.dni = CInt(Console.ReadLine())
            Console.Write("Ingrese Sueldo Básico: ")
            Oempleado.Sueldo_Basico = CDb1(Console.ReadLine())
            Console.Write("Ingrese Horas Extras: ")
            Oempleado.Horas_Extras = CInt(Console.ReadLine())
            Console.Write("Ingrese Monto por Hora Extra: ")
            Oempleado.Pago_Horas_Extras = CDb1(Console.ReadLine())
            Console.Write("Ingrese Porcentaje de descuento: ")
            Oempleado.descuento = CDb1(Console.ReadLine())
            Console.Write("Ingrese si es casado(si-no): ")
            Oempleado.casado = Console.ReadLine()
            opimp = imprimir()
            If opimp = 1 Then
                Oempleado.imprimir_datos_basicos()
            Else
                Oempleado.imprimir_datos_completos()
            End If
            Console.WriteLine()
            Console.WriteLine()
            Console.Write("Continua s/n: ")
            Do
                sn = Console.ReadLine()
                If sn.ToUpper <> "S" And sn.ToUpper <> "N" Then
                    Console.WriteLine("se debe ingresar S o N")
                    Console.ReadLine()
                End If
            Loop Until sn.ToUpper = "S" Or sn.ToUpper = "N"

            Catch ex As Exception
                Console.WriteLine(ex.Message)

        End Try
        Loop Until sn.ToUpper = "N"
        Console.ReadLine()
    End Sub

Private Function imprimir() As Integer
    Dim opcion As Integer
    Console.WriteLine()
    Console.WriteLine("IMPRESION")
    Console.WriteLine()
    Console.WriteLine("1 - Información Básica")
```

```

    Console.WriteLine("2 - Información Completa")
    Console.WriteLine()
    Console.Write("Opcion: ")
    Do
        opcion = CInt(Console.ReadLine())
    Loop Until opcion = 1 Or opcion = 2
    Return opcion
End Function
End Module

```

## CODIGO DE CLASE

```

Public Class Empleado
    Private _nombre As String
    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            If value.Length = 0 Then
                Throw New Exception("EL NOMBRE NO PUEDE ESTAR VACIO")
            Else
                _nombre = value
            End If
        End Set
    End Property
    Private _dni As Integer
    Public Property dni() As Integer
        Get
            Return _dni
        End Get
        Set(ByVal value As Integer)
            If value.ToString.Length < 8 Then
                Throw New Exception("El número de documento tiene que tener 8 digitos")
            Else
                _dni = value
            End If
        End Set
    End Property
    Private _Sueldo_Basico As Double
    Public Property Sueldo_Basico() As Double
        Get
            Return _Sueldo_Basico
        End Get
        Set(ByVal value As Double)
            If value < 0 Then
                Throw New Exception("El sueldo básico no puede ser negativo")
            Else
                _Sueldo_Basico = value
            End If
        End Set
    End Property

```

```

Private _Horas_Extras As Integer
Public Property Horas_Extras() As Integer
    Get
        Return _Horas_Extras
    End Get
    Set(ByVal value As Integer)
        If value < 0 Then
            Throw New Exception("Las horas extras no pueden ser negativas")
        Else
            _Horas_Extras = value
        End If
    End Set
End Property
Private _Pago_Horas_Extras As Double
Public Property Pago_Horas_Extras() As Double
    Get
        Return _Pago_Horas_Extras
    End Get
    Set(ByVal value As Double)
        If value < 0 Then
            Throw New Exception("EL valor de la hora extra no puede ser negativo")
        Else
            _Pago_Horas_Extras = value
        End If
    End Set
End Property
Private _descuento As Double
Public Property descuento() As Double
    Get
        Return _descuento
    End Get
    Set(ByVal value As Double)
        If value < 0 Then
            Throw New Exception("EL valor del descuento no puede ser negativo")
        Else
            _descuento = value
        End If
    End Set
End Property
Private _casado As String
Public Property casado() As String
    Get
        Return _casado
    End Get
    Set(ByVal value As String)
        If value.ToUpper = "SI" Or value.ToUpper = "NO" Then
            _casado = value
        Else
            Throw New Exception("Deberá ingresar si o no")
        End If
    End Set
End Property
Private _Numero_hijos As Integer
Public Property Numero_Hijos() As Integer

```

```

    Get
        Return _Numero_hijos
    End Get
    Set(ByVal value As Integer)
        If value < 0 Then
            Throw New Exception("El número de hijos no puede ser negativo")
        Else
            _Numero_hijos = value
        End If
    End Set
End Property

Private Function Monto_horas_Extras() As Double
    Return _Horas_Extras * Me.Pago_Horas_Extras
End Function

Private Function Monto_Sueldo_Bruto() As Double
    Return Me._Sueldo_Basico + Monto_horas_Extras()
End Function
Private Function Monto_retencion() As Double
    Dim retencion As Double = Me._descuento
    If Me.casado.ToUpper = "SI" Then
        retencion -= 2
    End If
    retencion -= Me._Numero_hijos
    Return Me.Monto_Sueldo_Bruto * retencion / 100
End Function
Private Function Monto_netto() As Double
    Return Me.Monto_Sueldo_Bruto - Me.Monto_retencion
End Function
Public Sub imprimir_datos_basicos(Optional ByVal datos As String = "básicos")
    Console.WriteLine()
    Console.WriteLine("Impresión de datos " & datos)
    Console.WriteLine()
    Console.WriteLine("Nombre: " & Me._nombre)
    Console.WriteLine("dni: " & Me._dni)
    Console.WriteLine("Casado: " & Me._casado)
    Console.WriteLine("Nº hijos: " & Me._Numero_hijos)
End Sub
Public Sub imprimir_datos_completos()
    Me.imprimir_datos_basicos("Completos")
    Console.WriteLine("Sueldo básico: " & Me._Sueldo_Basico)
    Console.WriteLine("Monto por horas extras: " & Monto_horas_Extras())
    Console.WriteLine("Monto Sueldo Bruto: " & Me.Monto_Sueldo_Bruto)
    Console.WriteLine("Monto de retencion: " & Me.Monto_retencion)
    Console.WriteLine("Sueldo Neto: " & Me.Monto_netto)

End Sub
End Class

```