

CODIGO CLIENTE

Module Module1

```
Sub Main()
    Dim oempleado As New Empleado
    Dim sn As String = "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("Continuas 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"
            If sn.ToUpper = "S" Then
                Console.Clear()
            End If
        Catch ex As Exception
            Console.WriteLine(ex.Message)
            Console.ReadLine()
        End Try
    Do While opimp = 1
    End Do
End Sub
```

```

        End Try

        Loop Until sn.ToUpper = "N"

    End Sub
    Private Function imprimir() As Integer
        Dim opcion As Integer
        Console.WriteLine()
        Console.Write("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 LA 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 NUMERO 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 BASICO NO PUEDE SER
NEGATIVO")
        Else
            _sueldo_basico = value
        End If
    End Set
End Property

Private _horas_extras As Double
Public Property horas_extras() As Double
    Get
        Return _horas_extras
    End Get
    Set(ByVal value As Double)
        If value < 0 Then
            Throw New Exception("LAS HORA EXTRAS NO PUEDE 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 Set
    End Property

```

```

        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.Length = 0 Then
            Throw New Exception("DEBERA INGRESA SI O NO")
        Else
            _casado = value
        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 NUMERO DE HIJOS NO PUEDE SER
NEGATIVO")
        Else
            _numero_hijos = value
        End If
    End Set
End Property

Private Function monto_horas_extras() As Double

```

```

        Return Me._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
        If retencion < 0 Then
            Return 0
        End If

        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 ByRef datos As String =
"bàsicos")
        Console.WriteLine()
        Console.WriteLine("Impresión de datos " & datos)
        Console.WriteLine()
        Console.WriteLine("Nombre: {0}", Me._nombre)
        Console.WriteLine("DNI: {0}", Me._dni)
        Console.WriteLine("Casado: {0}", Me._casado)
        Console.WriteLine("Nº de Hijos: {0}", Me._numero_hijos)
    End Sub
    Public Sub imprimir_datos_completos()
        Me.imprimir_datos_basicos("Completo")
        Console.WriteLine("Sueldo Básico: {0}", Me._sueldo_basico)
        Console.WriteLine("Monto por horas extras: {0}",
Me.monto_horas_extras)
        Console.WriteLine("Monto Sueldo Bruto: {0}",
Me.monto_sueldo_bruto)
        Console.WriteLine("Monto de Retencion: {0}", Me.monto_retencion)
        Console.WriteLine("Monto Netto: {0}", Me.monto_netto)
    End Sub
End Class

```