

Una de las características de la OOP, la encapsulación, establece que el código de una clase debe permanecer, siempre que sea posible, protegido de modificaciones no controladas del exterior (código cliente). Nuestra clase debe actuar como una especie de caja negra, que expone un interfaz para su uso, pero que no debe permitir el acceso a la implementación de dicho interfaz.

Supongamos que tenemos clase Empleado necesitamos crear un elemento para guardar el sueldo pagado, pero el importe del sueldo deberá estar entre un rango de valores en función de la categoría del empleado. Si la categoría es 1, el sueldo estará entre 1 y 200, mientras que si la categoría es 2, el sueldo podrá llegar hasta 300.

CODIGO CLIENTE

Module Module1

```
Sub Main()  
    Dim loEmpleado As Empleado  
    loEmpleado = New Empleado()  
        Try  
            loEmpleado.psNombre = "Pedro"  
            loEmpleado.Categoria = 2  
            loEmpleado.Sueldo = 150  
            Console.WriteLine("Empleado {0} - Categoria {1} - Sueldo {2}", loEmpleado.psNombre,  
loEmpleado.Categoria, loEmpleado.Sueldo)  
  
            loEmpleado.Sueldo = 175  
            Console.WriteLine("Empleado {0} - Categoria {1} - Sueldo {2}", loEmpleado.psNombre,  
loEmpleado.Categoria, loEmpleado.Sueldo)  
  
        Catch ex As Exception  
            Console.WriteLine(ex.Message)  
        End Try  
        Console.ReadLine()  
  
End Sub  
End Module
```

CODIGO CLASE

```
Public Class Empleado  
    Public psNombre As String  
        ' variables de propiedad  
    Private _Categoria As Integer  
    Private _bSueldo As Double  
        ' procedimientos de propiedad  
    Public Property Categoria() As Integer  
        Get  
            Return _Categoria  
        End Get  
        Set(ByVal Value As Integer)  
            _Categoria = Value  
        End Set  
    End Property  
    Public Property Sueldo() As Double  
        Get  
            Return _bSueldo  
        End Get  
    End Property  
End Class
```

```

' cuando asignamos el valor a esta propiedad,
' ejecutamos código de validación en el bloque Set
Set(ByVal Value As Double)

    If _Categoria < 1 Or _Categoria > 2 Then ' si la categoría es distinta de 1 y
2
        Throw New Exception("La categoría no corresponde con el sueldo")
        _bSueldo = 0
    ElseIf _Categoria = 1 And Value >= 200 Then ' si la categoría del empleado es
1 pero el sueldo supera 200

        ' Generando una excepción y asignar un cero
        Throw New Exception("La categoría no corresponde con el sueldo")
        _bSueldo = 0
    ElseIf _Categoria = 2 And Value <= 200 Or _Categoria = 2 And Value > 300 Then
' si la categoría del empleado es 2 pero el sueldo es menor o igual de 200 y mayor 300

        ' Generando una excepción y asignar un cero
        Throw New Exception("La categoría no corresponde con el sueldo")
        _bSueldo = 0
    Else
        ' si todo va bien, asignar el sueldo
        _bSueldo = Value
    End If
End Set
End Property
End Class

```