

## 1 Escribir números con letras II

Una vez que he escrito la clase [OperacionesEscribeNumeroConLetras] que encapsula todo el proceso de escribir un número en letras, queda por resolver todo el problema de su explotación práctica. Y de eso es de lo que se habla en este artículo.

### 1.1 Introducción.

En el capítulo I, he descrito y escrito el código necesario para escribir un número en letras. Ese código únicamente se encarga de escribir un número entero en letras y de nada más.

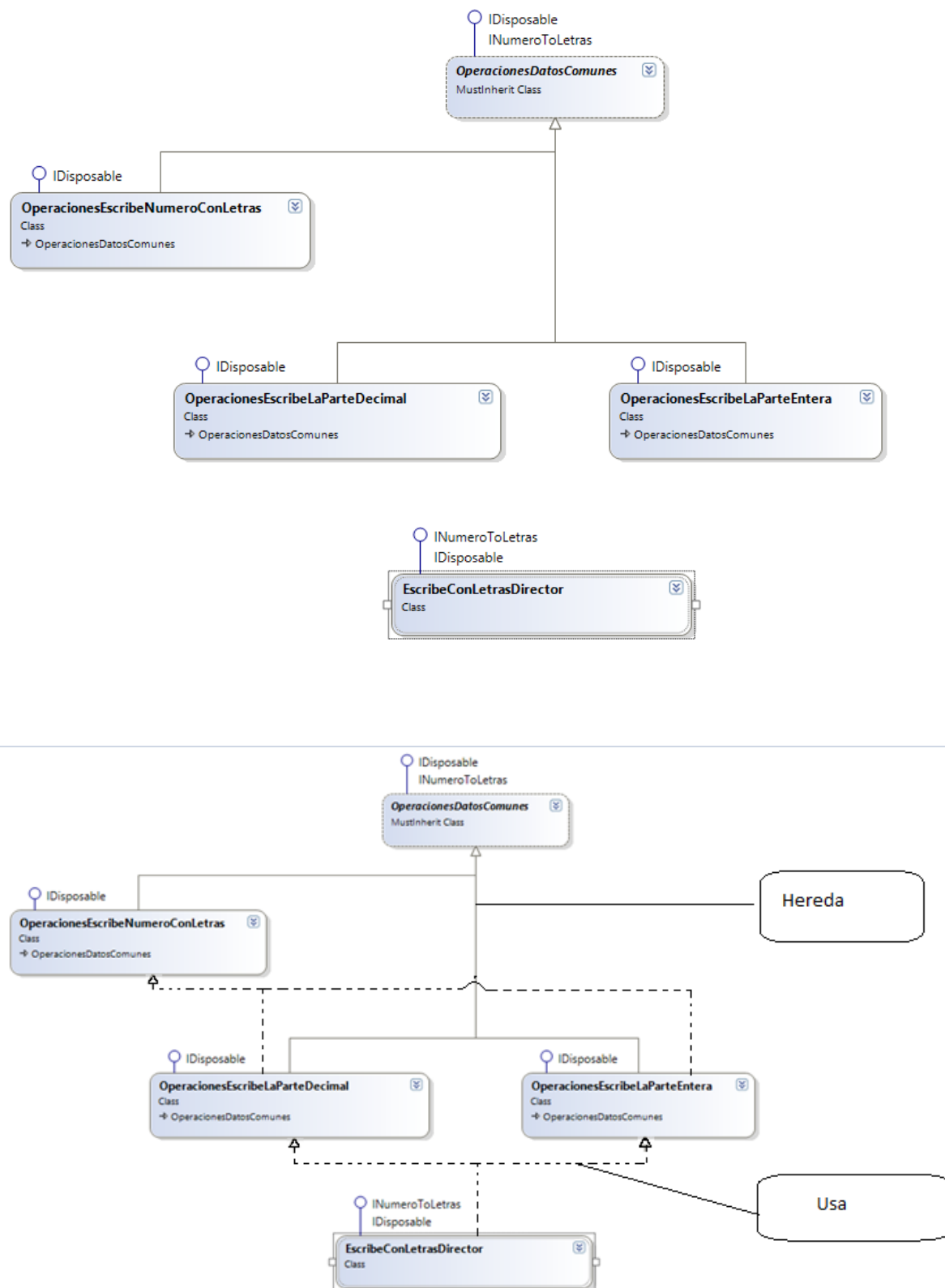
El problema surge cuando queremos ponerlo en producción porque en la vida real los números llevan decimales, y también pueden llevar un nombre que los califica (por ejemplo los euros). Si hacemos un estudio de la forma en la que se escriben los números vemos que si nos abstraemos de su posición se escriben igual la parte entera y la parte decimal, la única diferencia está en las unidades que caracterizan a cada parte Por ejemplo [euros] o [céntimos].

Con esta idea en la cabeza la solución es fácil: escribiremos las siguientes clases

- **[OperacionesDatosComunes]** Que contiene los datos comunes a todas las demás clases
- **[OperacionesEscribeNumeroConLetras]** que se encarga UNICAMENTE de escribir con letras los números
- **OperacionesEscribeLaParteEntera**, que se encarga de escribir SOLO la parte entera con sus unidades. Usa la clase contenida [OperacionesEscribeNumeroConLetras]
- **OperacionesEscribeLaParteDecimal**, que se encarga de escribir SOLO la parte decimal con sus unidades. Usa la clase contenida [OperacionesEscribeNumeroConLetras]
- **EscribeConLetrasDirector**, que se encarga de escribir con letras y con sus unidades correspondientes un número completo. Para hacer su trabajo usa las clases contenidas [OperacionesEscribeLaParteEntera] y [OperacionesEscribeLaParteDecimal]

## Escribir números con letras [DLL] (y II)

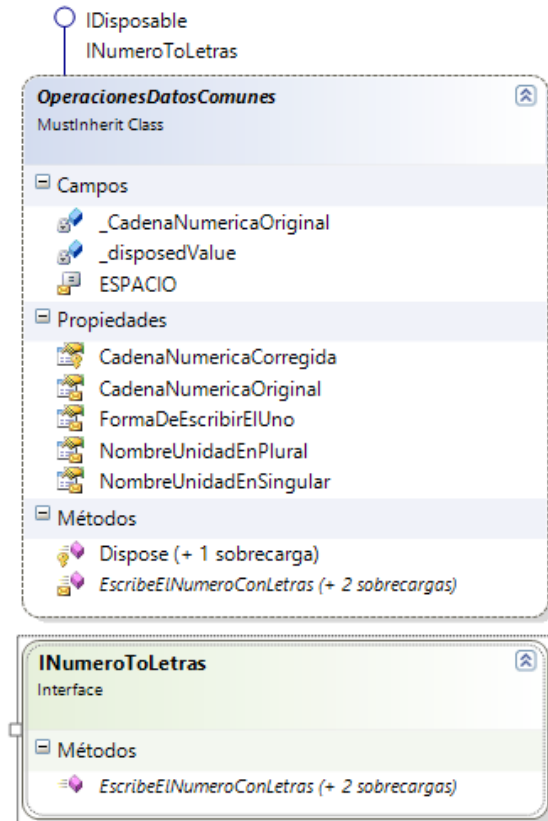
A continuación puedes ver el grafico de relaciones entre clases



## 1.2 Descripción de las clases:

### 1.2.1 [OperacionesDatosComunes]

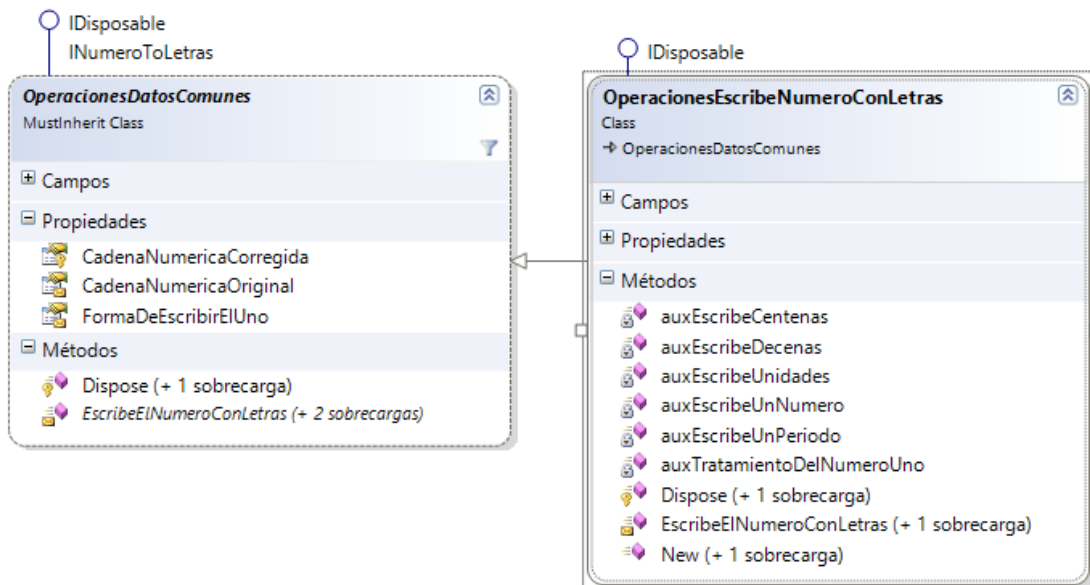
Clase que contiene los datos y propiedades comunes a todas las clases. Su objeto es evitar código duplicado



### 1.2.2 [OperacionesEscribeNumeroConLetras]

Es la clase que escribe con letras un número. Encapsula todas las operaciones para escribir un número [entero] y solo un número, en letras.

Esta clase resuelve toda la problemática de la ortografía, existente para escribir los números de forma correcta en español y además correctamente acentuados.

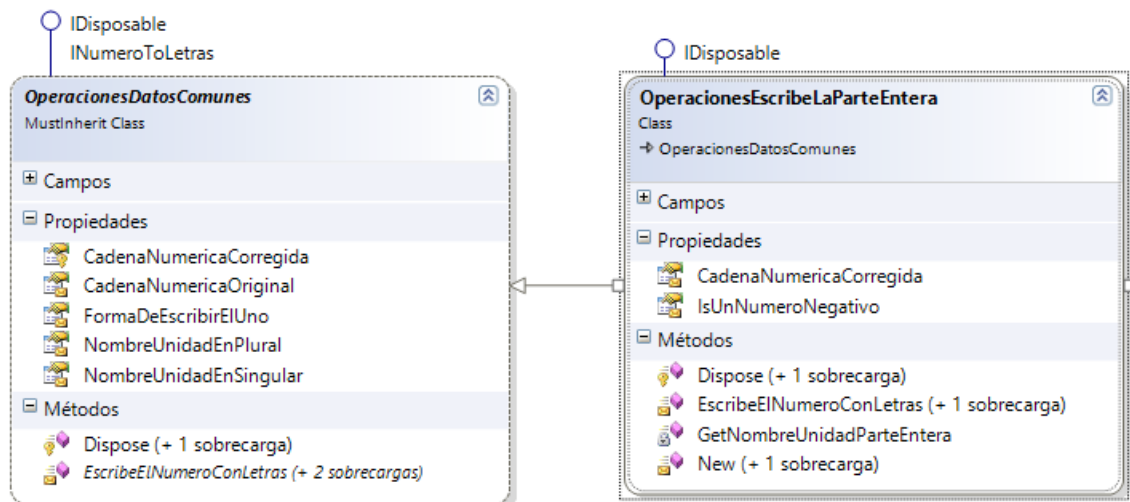


Esta es la clase que usa las reglas de escritura de los números en español dadas por la Real Academia de la lengua Española (RAE). Se encarga ÚNICAMENTE de escribir un número en letras

Las clases **[OperacionesEscribeLaParteEntera]**, y **[OperacionesEscribeLaParteDecimal]** llaman a esta clase para realizar el trabajo de escribir el número en letras

### 1.2.3 [OperacionesEscribeLaParteEntera].

Esta clase recibe de la clase director la parte entera del número, y la unidad que tiene que usar. Guarda el numero que corresponde a la parte entera, después corrige el numero quitando el [opcional] signo más (+) o menos (-) que pueda acompañar al número y quita los separadores decimales que pueda tener. También quita los ceros no significativos que haya a la izquierda del número (por ejemplo.: 000032 se guarda como 32). Para escribir el número en letras usa la clase contenida [OperacionesEscribeNumeroConLetras] que es la que se encarga de esa tarea.



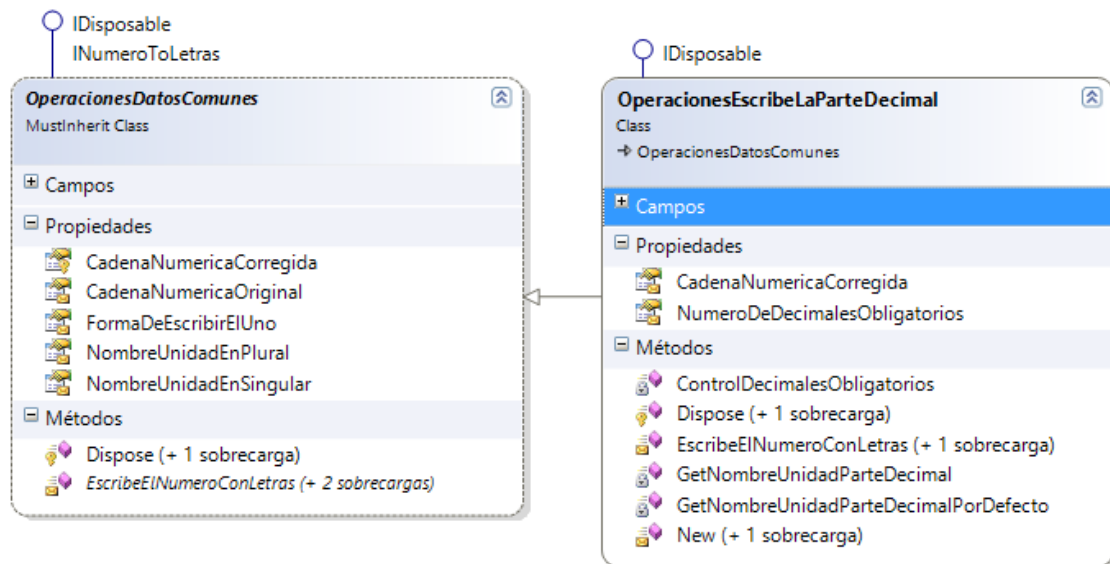
### 1.2.4 [OperacionesEscribeLaParteDecimal].

Esta clase recibe de la clase director la parte decimal del número (sin el separador decimal). Y su misión es: guardar el número que corresponde a la parte decimal, escribir en letras ese número y acompañarlo de su unidad correspondiente. También se encarga de comprobar que la cadena no tenga ningún carácter distinto de los dígitos. Y quita los ceros no significativos que hay a la derecha de la parte decimal de un numero (Por ejemplo 2,3400000 y 2,3400 es el mismo número 2,34).

También realiza dos tareas importantes.

- Controla el número de decimales que se imprimen de forma obligatoria. Un ejemplo; los euros siempre se escriben con dos decimales, pero la cantidad que se recibe puede tener cualquier numero de decimales y/o no tener ninguno. Una alternativa era dejar esa responsabilidad al cliente que usa la clase, pero al final me he decidido a situar aquí el código del control, entre otras cosas porque es más operativo y genérico.
- Si no existe unidades declaradas para la parte decimal, usa las denominaciones decimales estándar para calificarlas. La razón es que existían incoherencias al convertir los números. Por ejemplo [21,1] en letras es [veintiuno con uno], y el numero [21,01] que en letras TAMBIEN es [veintiuno con uno]. Para evitar esta indefinición he decidido que la función añadida de forma automática el nombre de la posición decimal ( únicamente en el caso de que el nombre que califica a la parte decimal no esté declarado,. Al final el resultado obtenido es el siguiente: [21,1] --> [Veintiuno con una décima] mientras que numero [21,01] se escribe como [ Veintiuno con una centésima]

## Escribir números con letras [DLL] (y II)



### 1.2.5 Nombres de las unidades decimales

Una observación

10(-1) debe leerse como 10 elevado a menos uno

10(-13) debe leerse como 10 elevado a menos trece

Lista de las primeras unidades decimales

10(-1) = 0,1 = "décima"

10(-2) = 0,01 = "centésima"

10(-3) = 0,001 = "milésima",

10(-4) = 0,0001 = "diezmilésima"

10(-5) = 0,00001 = "cienmilésima"

10(-6) = 0,000001 = "millonésima",

10(-7) = "diezmillonésima", 10(-8) = "cienmillonésima", 10(-9) = "milmillonésima",

10(-10) = "diezmilmillonésima", 10(-11) = "cienmilmillonésima", 10(-12) = "billonésima",

10(-13) = "diezbillonésima", 10(-14) = "cienbillonésima", 10(-15) = "milbillonésima",

"diezmilbillonésima", "cienmilbillonésima", "trillonésima",

"dieztrillonésima", "cientrillonésima", "miltrillonésima",

"diezmiltrillonésima", "cienmiltrillonésima", "cuatrillonésima",

"diezcuatrillonésima", "cienquatrillonésima", "milcuatrillonésima",

"diezmilcuatrillonésima", "cienmilcuatrillonésima", "quintillonésima",

"diezquintillonésima", "cienquintillonésima", "milquintillonésima",

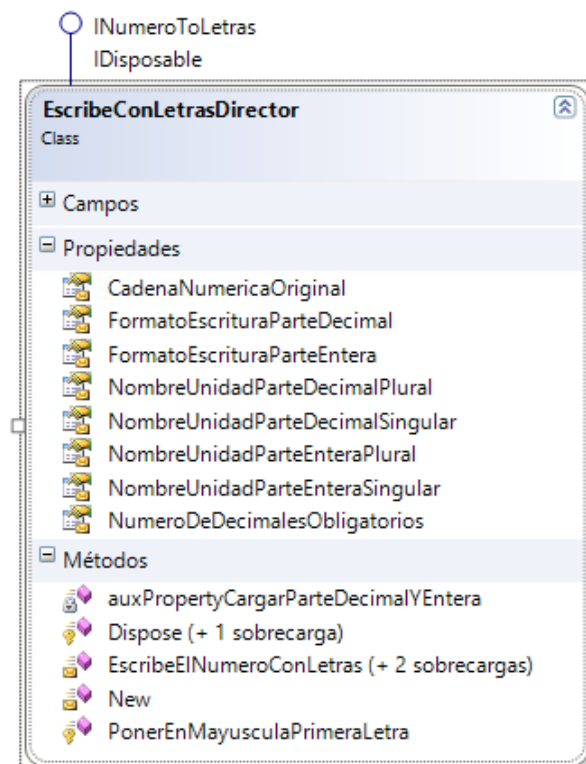
"diezmilquintillonésima", "cienmilquintillonésima", "sextillonésima",

## Escribir números con letras [DLL] (y II)

"diezsextilionésima",	"ciensextilionésima",	"milsextilionésima",
"diezmilsextilionésima",	"cienmilsextilionésima",	"septilionésima",
"diezseptilionésima",	"cienseptilionésima",	"milseptilionésima",
"diezmilseptilionésima",	"cienmilseptilionésima",	"octilionésima",
"diezoctilionésima",	"cienoctilionésima",	"miloctilionésima",
"diezmiloctilionésima",	"cienmiloctilionésima",	"novilionésima",
"dieznovilionésima",	"ciennovilionésima",	"milnovilionésima",
"diezmilnovilionésima",	"cienmilnovilionésima",	"vigetilionésima"

### 1.2.6 [EscribeConLetrasDirector]

Es la clase que recibe el número (parte entera y parte decimal) y las unidades que califican al número. Por ejemplo [45, 34], [euros], [céntimos] y devuelve [45 euros con 34 céntimos]. Para hacer esta tarea usa dos clases 'contenidas' que acabamos de ver: La clase [OperacionesEscribeLaParteEntera] que se encarga de tratar la parte entera, y la clase [OperacionesEscribeLaParteDecimal] que se encarga de tratar la parte decimal. Cada una de ellas trata su parte del número, lo convierten letras y lo devuelve a la clase director que es la que se encarga de montar la cadena final que es la que se entrega al usuario.



### 1.3 La clase [GenericaEscribeConLetras]

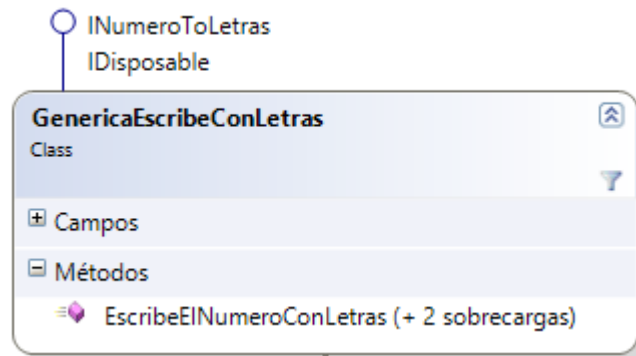
Todas las clases relacionadas anteriormente son clases cuyo alcance es Friend, es decir no se ven fuera de la DLL, por eso he escrito la clase [GenericaEscribeConLetras.vb] que es publica y usa la como clase contenida clase [EscribeConLetrasDirector] para hacer el trabajo. También implementa la interfaz [INumeroToLetras] para poder usarla como interfaz en lugar de cómo clase instanciada.

En esta clase se recibe a través del constructor todos los datos que se necesitan para trabajar que son:

- El sustantivo en SINGULAR que califica la parte ENTERA Se espera un valor [string.empty] o bien, una cadena que contiene un sustantivo (en singular) que califica la parte entera del número. Por ejemplo Euro, patata,
- El sustantivo en PLURAL que califica la parte ENTERA Se espera un valor [string.empty] o bien, una cadena que contiene un sustantivo (en plural) Por ejemplo Euros, patatas
- El formato de escritura del valor [uno] en la parte ENTERA. El sustantivo que califica a la parte entera del número (si existe) ¿es del género masculino o femenino? Por ejemplo, euro(s) es masculino, patata(s) es femenina
- El sustantivo en SINGULAR que califica la parte DECIMAL Se espera un valor [string.empty] o bien, una cadena que contiene un sustantivo (en singular) que califica la parte decimal del número. Por ejemplo céntimo, patatilla
- El sustantivo en PLURAL que califica la parte DECIMAL. Se espera un valor [string.empty] o bien, una cadena que contiene un sustantivo (en plural) Por ejemplo. Por ejemplo céntimos, patatillas
- El formato de escritura del valor [uno] en la parte DECIMAL. El sustantivo que califica a la parte decimal del número (si existe) ¿es del género masculino o femenino? Por ejemplo, céntimo(s) es masculino, patatilla(s) es femenina
- El numero de decimales que obligatoriamente debe tener el número. Por ejemplo los EUROS, SIEMPRE tienen DOS (2) decimales. Su valor por defecto= [-1] que significa que no se controla este valor, Es decir, el numero puede tener (o no) cualquier numero de decimales

```
Public Sub New(ByVal pNombreUnidadParteEnteraSingular As String,
               ByVal pNombreUnidadParteEnteraPlural As String,
               ByVal pFormatoEscrituraParteEntera As FormatoEscrituraDelUno,
               ByVal pNombreUnidadParteDecimalSingular As String,
               ByVal pNombreUnidadParteDecimalPlural As String,
               ByVal pFormatoEscrituraParteDecimal As FormatoEscrituraDelUno,
               ByVal pNumeroDeDecimalesObligatorios As Integer)
```

Escribir números con letras [DLL] (y II)



También contiene las funciones que se encargan de escribir el número en letras, y para ello delegan el trabajo en la clase contenida [EscribeConLetrasDirector]

Por ejemplo:

- Primero declarar el objeto contenido

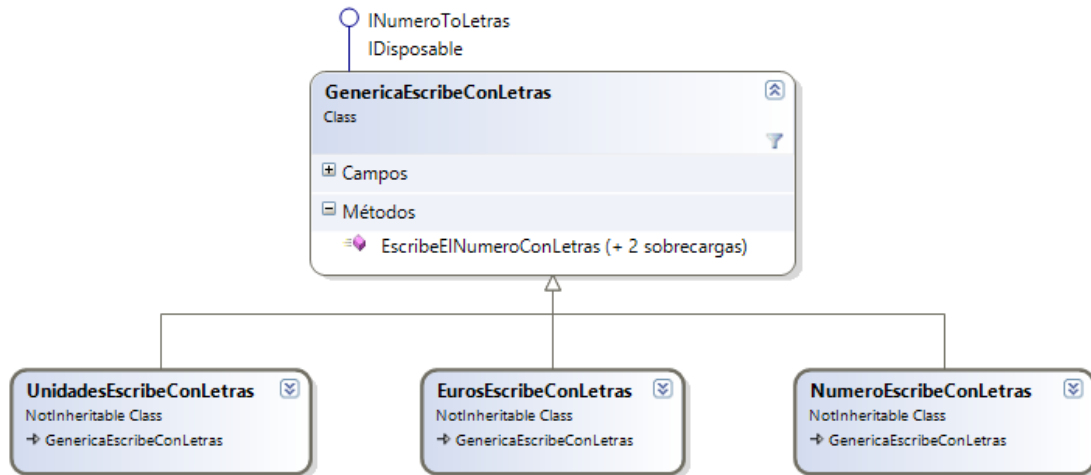
```
''' <summary>
'''     El objeto contenido que encapsula toda la
'''     problemática de convertir un numero en letras
''' </summary>
''' <remarks>
'''     Observación
'''     Lo uso como objeto contenido porque así evito que se vea fuera de la DLL
''' </remarks>
Private objEscrivor As EscribeConLetrasDirector
```

- Segundo: en el constructor cargar los datos en el objeto contenido ( no se muestra el código)
- Tercero Usar la clase contenida para hacer el trabajo

```
'<summary>
'''   Escribe en letras un número
'''   (P.E. 45,23 -> Cuarenta y cinco con veintitrés)
'''</summary>
'''   <param name="pCadenaNumerica">
'''       <para>La cadena que contiene el numero. </para>
'''       <para>Es responsabilidad del llamador asegurarse de que
'''           la cadena es un numero bien formado,
'''           escrito en formato numérico Español es decir:</para>
'''       <para>El separador decimal es la coma [,]. </para>
'''       <para>El separador de los millares es o bien
'''           el punto [.] , o bien el espacio [' ']. </para>
'''       <para>La cadena debe contener únicamente números
'''           (no están permitidas las letras),</para>
'''       <para>Opcionalmente puede llevar un signo más [+] o menos [-]
'''           para indicar que el número es positivo o negativo.</para>
'''   </param>
'''   <returns>
'''       <para>Una cadena que contiene escrito en letras el numero pasado</para>
'''       <para>si el numero es cero (0) devuelve una cadena con el valor [cero] </para>
'''       <para>si el numero es una cadena vacía [Empty] se devuelve [Empty]</para>
'''   </returns>
Public Function EscribeElNumeroConLetras(pCadenaNumerica As String) As String _
    Implements INumeroToLetras.EscribeElNumeroConLetras
    Return _objEscritor.EscribeElNumeroConLetras(pCadenaNumerica)
End Function
```

## Escribir números con letras [DLL] (y II)

La DLL tiene predefinidas tres clases que hacen un trabajo específico. En el diagrama siguiente se puede ver la cadena de herencias.



Las clases son:

- **[NumeroEscribeConLetras]** Que escribe un número con letras sin usar ninguna unidad. Observa que siempre escribe la unidad decimal por defecto. Por ejemplo [(21,021 ) = Veintiuno con veintiuna milésimas]
- **[UnidadesEscribeConLetras]** Que escribe un número con letras usando como unidad la palabra [unidad(es)]. Observa que siempre escribe la unidad decimal por defecto. Por ejemplo [(21,021 ) = Veintiuna unidades con veintiuna milésimas ]
- **[EurosEscribeConLetras]** que escribe la cantidad como si fueran euros. Por ejemplo [(21,021) = Veintiún euros con dos céntimos]

### 1.3.1 A modo de ejemplo, la clase [EurosEscribeConLetras]

```
''' <summary>
'''     Escribe en letras cadenas con números que representan euros
''' </summary>
''' <remarks>
'''     <a href="http://www.hispanoteca.eu/Foro-preguntas/ARCHIVO-Foro/Euro%20-%20euros.htm">
'''         Hispanoteca - Como se escribe 'euro'
'''     </a>
''' </remarks>
Public NotInheritable Class EurosEscribeConLetras
    Inherits GenericaEscribeConLetras

    ''' <summary>
    '''     Constructor
    ''' </summary>
    Public Sub New()
        MyBase.New(
            "euro", "euros", FormatoEscrituraDelUno.UnConSustantivoMasculino,
            "céntimo", "céntimos", FormatoEscrituraDelUno.UnConSustantivoMasculino, 2)
    End Sub
End Class
```

## 1.4 Forma de usar la DLL

### Una observación

Esta función esta copiada de un formulario que tiene dos cuadros de texto:

- TextBoxNumero: que es el cuadro de texto donde el usuario introduce el numero
- TextBoxLetras: que es el cuadro de texto donde se escribe el numero en letras

Prueba de escritura de numeros

Numero

32,41

☒ Solo numeros ☐ Unidad ☐ Euros

Convertir Cancelar

Traduccion a palabras

(1) Treinta y dos con cuarenta y una centésimas  
(2) Treinta y dos unidades con cuarenta y una centésimas  
(3) Treinta y dos euros con cuarenta y un céntimos  
(4) Treinta y dos manzanas con cuarenta y una centimanzanas

### 1.4.1 Con la interface

```

'/* Forma de usar la interface desde código */
Private Sub TrabajoConInterfaces()
    Try

        ' Evitar un problema tonto con el cuadro de texto
        If Me.TextBoxLetras Is Nothing Then Exit Sub

        Dim interfaceConvertidor As NumerosToLetrasDll.INumeroToLetras = Nothing

        If Me.RadioButtonSoloNumeros.IsChecked = True Then
            interfaceConvertidor = New NumerosToLetrasDll.NumeroEscribeConLetras
        End If
        If Me.RadioButtonUnidad.IsChecked = True Then
            interfaceConvertidor = New NumerosToLetrasDll.UnidadesEscribeConLetras
        End If
        If Me.RadioButtonEuros.IsChecked = True Then
            interfaceConvertidor = New NumerosToLetrasDll.EurosEscribeConLetras
        End If

        Me.TextBoxLetras.AppendText(Environment.NewLine)
        Me.TextBoxLetras.AppendText("(5) " &
            interfaceConvertidor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))

        ' -----
        interfaceConvertidor = New NumerosToLetrasDll.GenericaEscribeConLetras(
            pNombreUnidadParteEnteraSingular:="manzana",
            pNombreUnidadParteEnteraPlural:="manzanas",
            pFormatoEscrituraParteEntera:=.
                FormatoEscrituraDelUno.UnaConSustantivoFemenino,
            pNombreUnidadParteDecimalSingular:="centimanzana",
            pNombreUnidadParteDecimalPlural:="centimanzanas",
            pFormatoEscrituraParteDecimal:=.
                FormatoEscrituraDelUno.UnaConSustantivoFemenino,
            pNumeroDeDecimalesObligatorios:=-1)

        Me.TextBoxLetras.AppendText(Environment.NewLine)
        Me.TextBoxLetras.AppendText("(6) " &
            interfaceConvertidor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))

    Catch ex As Exception
        If Not (Me.TextBoxLetras Is Nothing) Then
            Me.TextBoxLetras.Text = ex.Message
        End If
    End Try
End Sub

```

## 1.4.2 Instanciando las clases

```
Private Sub trabajoConClases()  
    Try  
        ' Evitar un problema tonto con el cuadro de texto  
        If Me.TextBoxLetras Is Nothing Then Exit Sub  
  
        Me.TextBoxLetras.Clear()  
  
        ' escribe doce con treinta y dos centesimas  
        Using objEscitor = New NumerosToLetrasDll.NumeroEscribeConLetras  
            'Me.TextBoxLetras.AppendText(Environment.NewLine)  
            Me.TextBoxLetras.AppendText("(1) " &  
                objEscitor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))  
        End Using  
  
        ' escribe doce unidades con treinta y dos centesimas  
        Using objEscitor = New NumerosToLetrasDll.UnidadesEscribeConLetras  
            Me.TextBoxLetras.AppendText(Environment.NewLine)  
            Me.TextBoxLetras.AppendText("(2) " &  
                objEscitor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))  
        End Using  
  
        ' escribe doce euros con veinte centimos  
        Using objEscitor = New NumerosToLetrasDll.EurosEscribeConLetras  
            Me.TextBoxLetras.AppendText(Environment.NewLine)  
            Me.TextBoxLetras.AppendText("(3) " &  
                objEscitor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))  
        End Using  
  
        ' la clase genérica para cualquier unidad  
        Using objEscitor = New NumerosToLetrasDll.GenericaEscribeConLetras(  
            pNombreUnidadParteEnteraSingular:="manzana",  
            pNombreUnidadParteEnteraPlural:="manzanas",  
            pFormatoEscrituraParteEntera:=  
                FormatoEscrituraDelUno.UnaConSustantivoFemenino,  
            pNombreUnidadParteDecimalSingular:="centimanzana",  
            pNombreUnidadParteDecimalPlural:="centimanzanas",  
            pFormatoEscrituraParteDecimal:=  
                FormatoEscrituraDelUno.UnaConSustantivoFemenino,  
            pNumeroDeDecimalesObligatorios:=-1)  
  
            Me.TextBoxLetras.AppendText(Environment.NewLine)  
            Me.TextBoxLetras.AppendText("(4) " &  
                objEscitor.EscribeElNumeroConLetras(Me.TextBoxNumero.Text))  
        End Using  
  
        Catch ex As Exception  
            If Not (Me.TextBoxLetras Is Nothing) Then  
                Me.TextBoxLetras.Text = ex.Message  
            End If  
        End Try  
    End Sub
```