

Vamos a desarrollar un ejemplo de un cambio de divisas, con la siguiente interfaz

Como hemos visto en el Ejercicio1 y en el Ejercicio2, el diseño correría a cargo de una hoja de estilos CSS que en este ejemplo no hemos utilizado.

1

La forma más fácil de crear nuestra aplicación sería utilizar el método **onClick** de cada botón. Nuestro código quedaría:

Default.aspx

```
<body style="height: 233px">
  <form id="form1" runat="server">
    <div>

      <asp:Label ID="lbleuros" runat="server" Font-Bold="True" Text="Euros:"
      Tooltip="Introduzca los euros a cambiar" Width="100px"></asp:Label>

      <asp:TextBox ID="txteuros" runat="server"></asp:TextBox>

      <asp:Label ID="lblecambio" runat="server"></asp:Label>
      <asp:TextBox ID="txtcambio" runat="server" ReadOnly="True"></asp:TextBox>

      <asp:Button ID="btnDOLARCAN" runat="server" AccessKey="a"
      OnClick="CalcularDOLARCAN" Text="Dolar CAN" width="93px" />

      <asp:Button ID="btndolarAUS" runat="server" height="26px"
      OnClick="CalcularDOLARAUS" Text="US Dolar" />

      <asp:Button ID="btnPESOARG" runat="server" height="26px"
      OnClick="CalcularPESOARG" Text="Peso ARG" />

      <asp:Button ID="btnDolarUSA" runat="server" OnClick="CalcularDOLARUSA"
      Text="Dolar AUS" />
    </div>

  </form>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    private const double dUSA = 1.35;
    private const double dAUS = 1.43;
    private const double dCAN = 1.39;
    private const double pARG = 7.78;
    double monedas;*/

    protected void CalcularDOLARAUUS(object sender, EventArgs e)
    {
        monedas = Convert.ToDouble(txteuros.Text);
        txtcambio.Text = (monedas * dAUS).ToString("g");
    }
    protected void CalcularDOLARCAN(object sender, EventArgs e)
    {
        monedas = Convert.ToDouble(txteuros.Text);
        txtcambio.Text = (monedas * dCAN).ToString("g");
    }
    protected void CalcularDOLARUSA(object sender, EventArgs e)
    {
        monedas = Convert.ToDouble(txteuros.Text);
        txtcambio.Text = (monedas * dUSA).ToString("g");
    }
    protected void CalcularPESOARG(object sender, EventArgs e)
    {
        monedas = Convert.ToDouble(txteuros.Text);
        txtcambio.Text = (monedas * pARG).ToString("g");
    }
}
```

2

La segunda forma es llamando desde cada uno de los botones a través del método **OnClick** al mismo procedimiento, dado que todos hacen las mismas operaciones y lo único que cambia es el valor por el que multiplicamos la cantidad:

Default.aspx

```
<body style="height: 233px">
    <form id="form1" runat="server">
        <div>

            <asp:Label ID="lbleuros" runat="server" Font-Bold="True" Text="Euros:"
            Tooltip="Introduzca los euros a cambiar" Width="100px"></asp:Label>

            <asp:TextBox ID="txteuros" runat="server"></asp:TextBox>

            <asp:Label ID="lblcambio" runat="server"></asp:Label>
            <asp:TextBox ID="txtcambio" runat="server" ReadOnly="True"></asp:TextBox>
```

```

        <asp:Button ID="btnDOLARCAN" runat="server" AccessKey="a"
        OnClick="Calcular" Text="Dolar CAN" width="93px" />

        <asp:Button ID="btndolarAUS" runat="server" height="26px"
        OnClick="Calcular" Text="US Dolar" />

        <asp:Button ID="btnPESOARG" runat="server" height="26px"
        OnClick="Calcular" Text="Peso ARG" />

        <asp:Button ID="btnDolarUSA" runat="server" OnClick="Calcular"
        Text="Dolar AUS" />
    </div>

</form>

```

Default.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Calcular(object sender, EventArgs e)
    {
        monedas = Convert.ToDouble(txtteuros.Text);
        Button btncambio=(Button)sender;
        double cambio=1;
        switch (btncambio.ID){
            case "btnDOLARCAN":
                cambio=1.39;
                break;
            case "btndOLARAUS":
                cambio=1.43;
                break;
            case "btnDOLARUSA":
                cambio=1.35;
                break;
            case "btnPESOARG":
                cambio=7.78;
                break;
        }
        txtcambio.Text = (monedas * cambio).ToString("g");
    }
}

```

3

En esta tercera opción vamos a trabajar con las propiedades que tienen los botones (bien sean de comandos (CommandButton), de enlace (LinkButton) y de imagen (ImageButton) **CommandName** y **CommandArgument** y jugaremos con el método **OnCommand** en vez del método **OnClick**.

Este método **OnCommand** se ejecuta también cuando pulsamos el botón, pero la sintaxis del método es :

```
protected void nombreprocedimiento(object sender, CommandEventArgs e)
{
}
}
```

Si nos fijamos en la sintaxis de la declaración, **el parámetro e** es de tipo **CommandEventArgs** en vez de **EventArgs** (como lo era el parámetro del procedimiento correspondiente al evento **Click**). Esto hace que desde este parámetro (**e**) podamos acceder a las propiedades **CommandName** y **CommandArgument** del elemento (botón) que hayamos pulsado. Por tanto, si introducimos los **nombres de las monedas** en la propiedad **CommandName** de cada botón y los **cambios** en la propiedad **CommandArgument** de cada botón, podríamos acceder a estos valores desde el parámetro **e** de la siguiente forma:

e.CommandName

e.CommandArgument

Default.aspx

```
<body style="height: 233px">
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="lbleuros" runat="server" Font-Bold="True" Text="Euros:"
        ToolTip="Introduzca los euros a cambiar" Width="100px"></asp:Label>

      <asp:TextBox ID="txteuros" runat="server"></asp:TextBox>

      <asp:Label ID="lblcambio" runat="server"></asp:Label>
      <asp:TextBox ID="txtcambio" runat="server" ReadOnly="True"></asp:TextBox>

      <asp:Button ID="Button2" runat="server" OnCommand="calcular"
        Text="Dolar AUS" CommandArgument="1,43" CommandName="Dolares Australianos"
        />

      <asp:Button ID="Button1" runat="server" height="26px" OnCommand="calcular"
        Text="US Dolar" CommandArgument="1,35" CommandName="Dolares Americanos"
        />
      <asp:Button ID="Button3" runat="server" AccessKey="a" OnCommand="calcular"
        Text="Dolar CAN" width="93px" CommandArgument="1,39" CommandName="Dolares
        Canadienses" />
      <asp:Button ID="Button4" runat="server" height="26px" OnCommand="calcular"
        Text="Peso ARG" CommandArgument="7,78" CommandName="Pesos Argentinos" />
    </div>
  </form>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{

    protected void calcular(object sender, CommandEventArgs e)
    {
        double cambio=Convert.ToDouble(e.CommandArgument);
        double moneda=Double.Parse(txtteuros.Text);
        lblcambio.Text = e.CommandName;
        double resultado= (monedas * cambio);
        txtcambio.Text = resultado.ToString();
    }
}
```

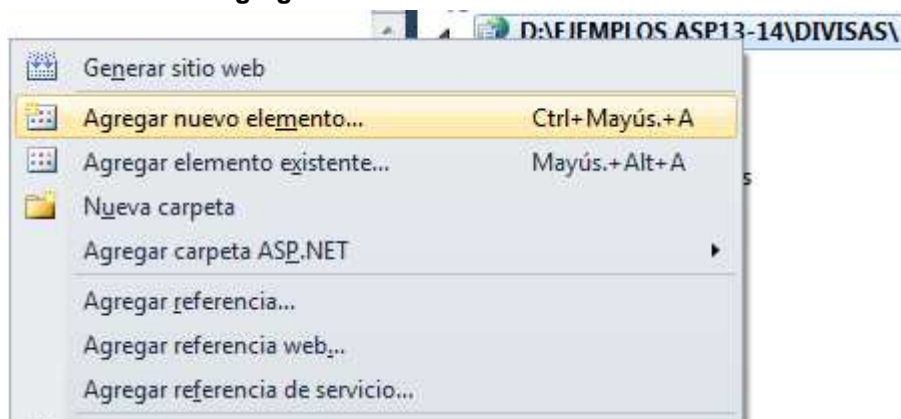
4

En la tercera for

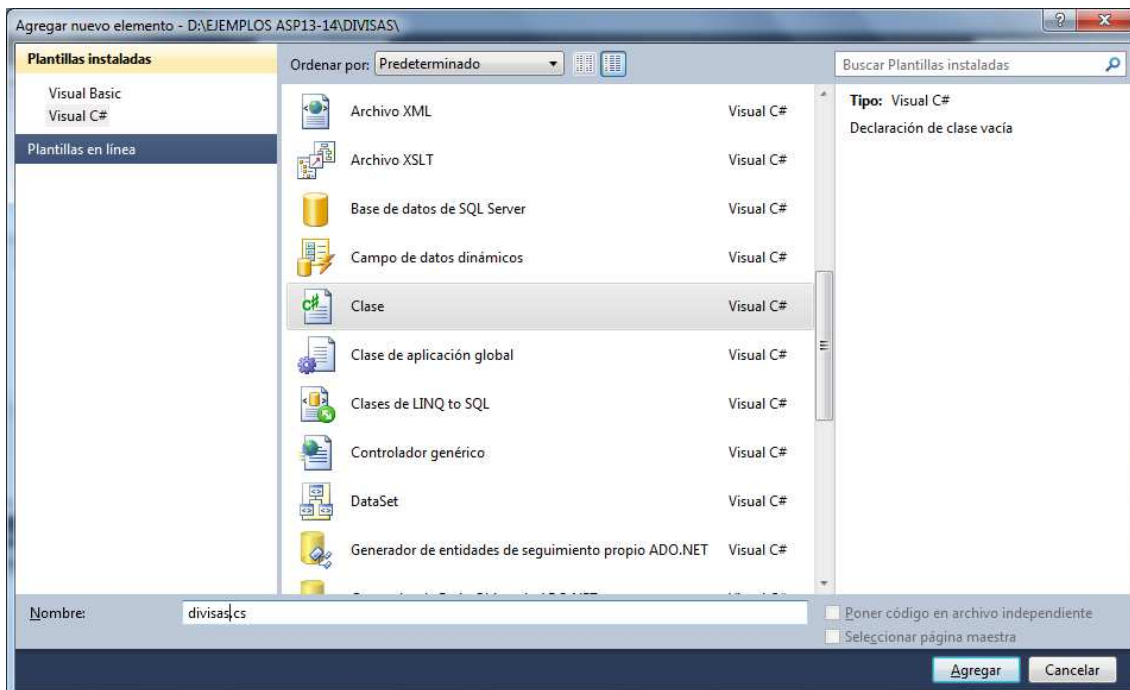
o derivadas (abstract sealed) sino tan sólo a proporcionar ciertos métodos estáticos.

Una clase estática es básicamente igual que una clase no estática, pero existe una diferencia: como ya hemos dicho **no se pueden crear instancias** de una clase estática. En otras palabras, no puede utilizar la palabra clave **new** para crear una variable del tipo clase. Dado que no hay ninguna variable de instancia, **el acceso a los miembros de una clase estática se realiza mediante el propio nombre de clase.**

Crearemos la clase. Para ello Pulsaremos sobre el proyecto con el botón derecho y seleccionaremos **Agregar nuevo elemento...**



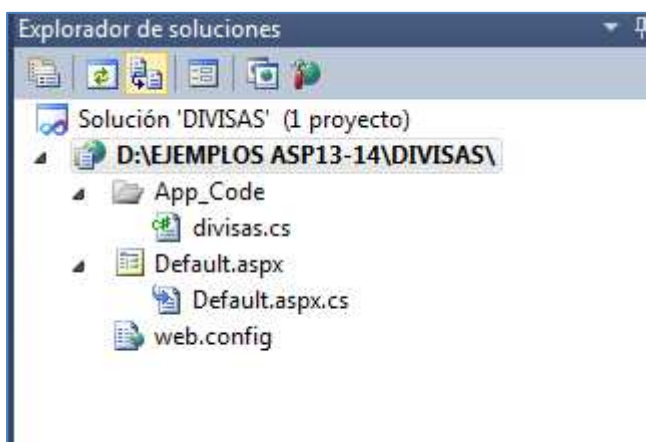
Seleccionaremos Clase y le daremos el nombre deseado. En nuestro caso **divisas.cs**.



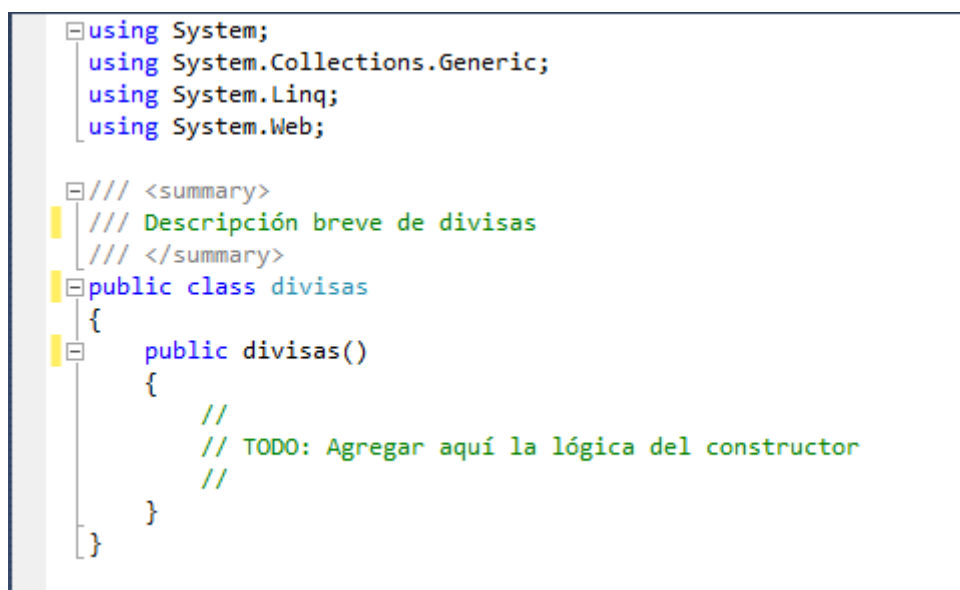
Nos aparecerá el siguiente mensaje en el que se nos informa que las clases deben ir dentro de la carpeta **App_Code**



Pulsaremos **Sí** y se nos mostrará el siguiente esquema en el **explorador de soluciones**.

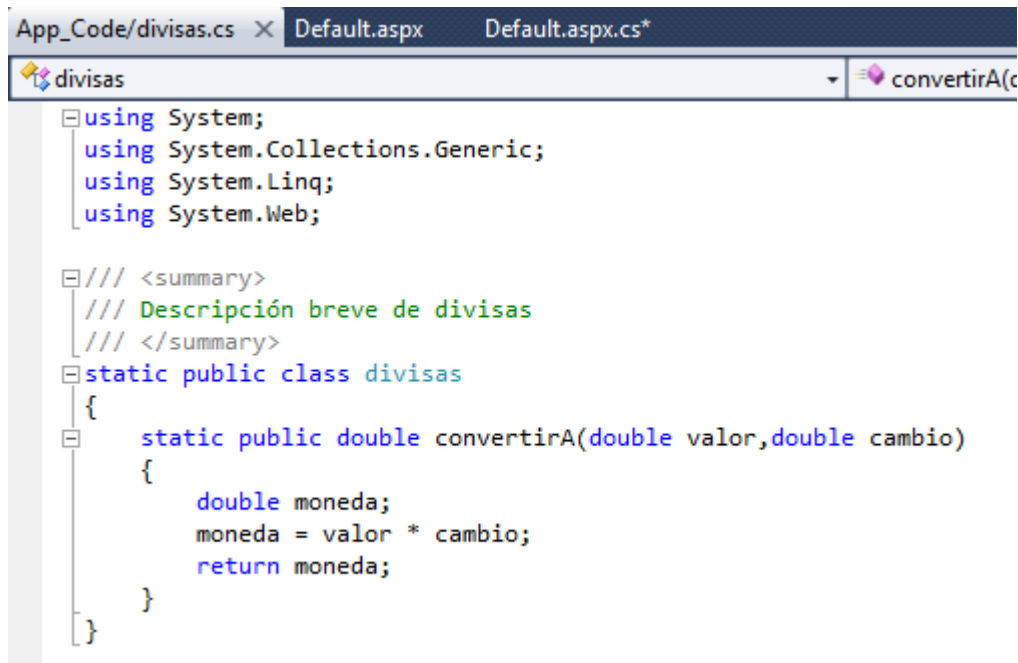


Y en el **área de trabajo** se nos mostrará el **esqueleto de la clase** divisas.



La completaremos. En nuestro caso es una **clase estática** por tanto por delante de la clase debemos poner el modificador **static** y **delante de su método también**, dado que todos los métodos de una clase estática, deben ser estáticos.

Clase estática Divisas.cs



```
App_Code/divisas.cs x Default.aspx Default.aspx.cs*
divisas
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Descripción breve de divisas
/// </summary>
static public class divisas
{
    static public double convertirA(double valor,double cambio)
    {
        double moneda;
        moneda = valor * cambio;
        return moneda;
    }
}
```

Default.aspx

```
<body style="height: 233px">
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="lbleuros" runat="server" Font-Bold="True" Text="Euros:"
        ToolTip="Introduzca los euros a cambiar" Width="100px"></asp:Label>

      <asp:TextBox ID="txteuros" runat="server"></asp:TextBox>

      <asp:Label ID="lblcambio" runat="server"></asp:Label>
      <asp:TextBox ID="txtcambio" runat="server" ReadOnly="True"></asp:TextBox>

      <asp:Button ID="Button2" runat="server" OnCommand="calcular"
        Text="Dolar AUS" CommandArgument="1,43" CommandName="Dolares Australianos"
        />

      <asp:Button ID="Button1" runat="server" height="26px" OnCommand="calcular"
        Text="US Dolar" CommandArgument="1,35" CommandName="Dolares Americanos"
        />
      <asp:Button ID="Button3" runat="server" AccessKey="a" OnCommand="calcular"
        Text="Dolar CAN" width="93px" CommandArgument="1,39" CommandName="Dolares
        Canadienses" />
      <asp:Button ID="Button4" runat="server" height="26px" OnCommand="calcular"
        Text="Peso ARG" CommandArgument="7,78" CommandName="Pesos Argentinos" />
    </div>
  </form>
```


Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

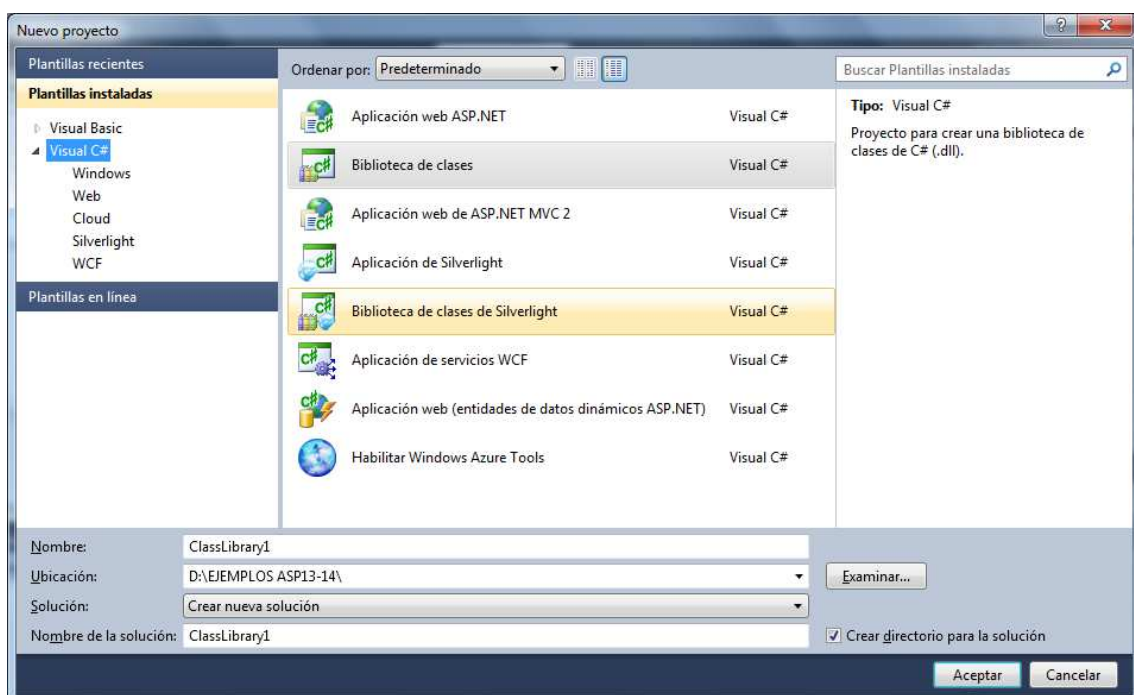
public partial class _Default : System.Web.UI.Page
{
    protected void calcular(object sender, CommandEventArgs e)
    {
        double cambio=Convert.ToDouble(e.CommandArgument);
        double moneda=Double.Parse(txteuros.Text);
        lblcambio.Text = e.CommandName;
        double resultado=divisas.convertirA(moneda,cambio);
        txtcambio.Text = resultado.ToString();
    }
}
```

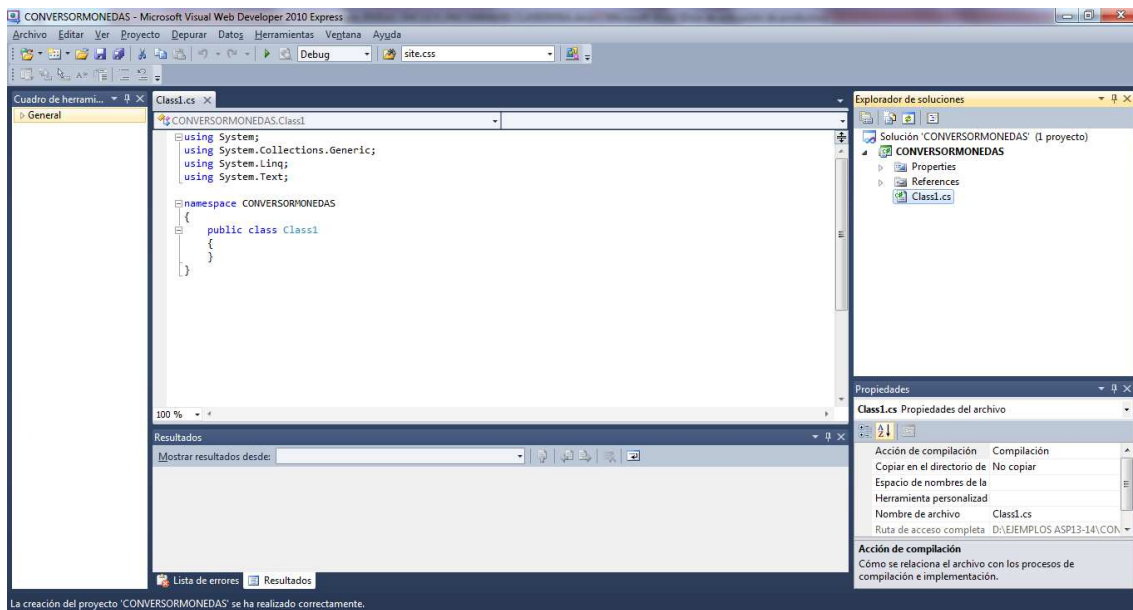
5

Esta última opción es igual a la anterior, solo que en vez de crear la clase en el propio sitio web. Vamos a crear **una biblioteca con la clase** y la vamos a **referenciar** en nuestro **sitio Web** para poder **acceder a sus clases**.

Para crear la librería hay que Seleccionar **Nuevo Proyecto** ➤ **Biblioteca de Clases**

Pondremos el nombre y será el que se coloque como Espacio de Nombres, por ejemplo **CONVERSORMONEDAS**.





Aquí escribiremos la clase del punto 4.

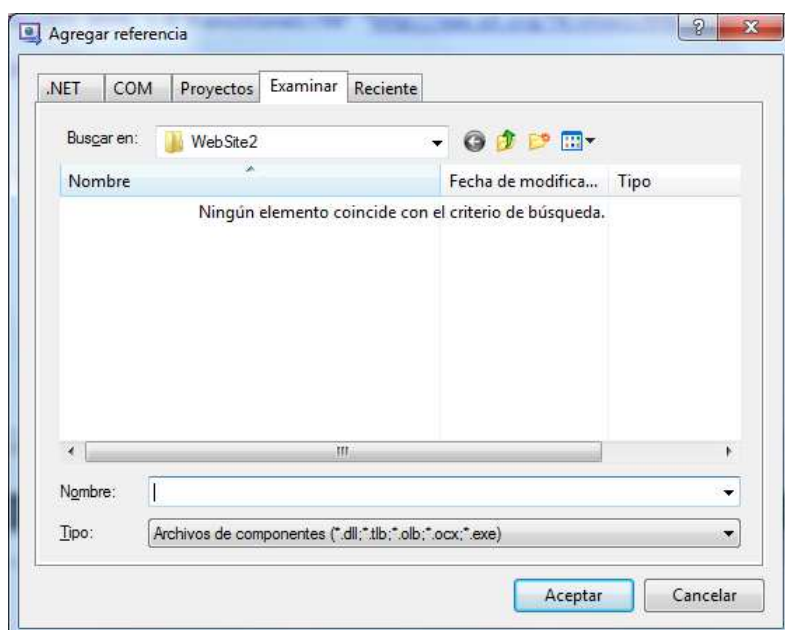
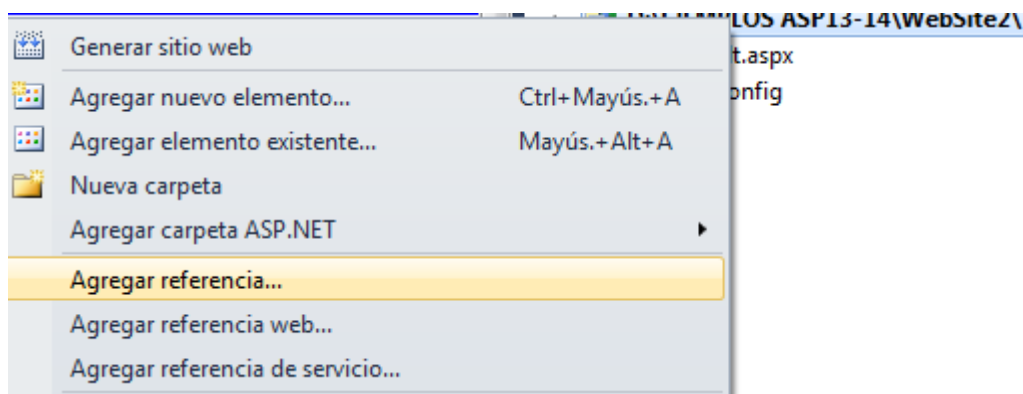
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CONVERSORMONEDAS
{
    static public class divisas
    {
        static public double convertirA(double valor, double cambio)
        {
            double moneda;
            moneda = valor * cambio;
            return moneda;
        }
    }
}
```

Pulsando **Depurar** ➤ **Generar** **ConversorMonedas** generaremos la **dll** (**ConversorMonedas.dll**) en el directorio **Bin/Debug** que se encuentra dentro de la carpeta en la que hemos creado el proyecto.

Posteriormente, crearemos nuestro Sitio Web, pulsando **Nuevo Sitio Web** ➤ **Sitio Web Vacío** y añadiremos como siempre nuestro **Web Form**, en donde diseñaremos el formulario que hemos visto al inicio del ejercicio.

Ahora tenemos que **añadir la referencia a nuestra biblioteca**, para ello pulsaremos con el **botón derecho** encima del **proyecto** en el **Explorador de Soluciones** y pulsaremos **Añadir Referencia**.



En la solapa **Examinar** buscaremos la **dll** correspondiente. Y se agregará al sitio Web. Una vez agregada la **referencia** ya podemos hacer uso de nuestra clase. Si queremos podemos importar el **espacio de nombres** (instrucción **using**), para poder acceder directamente a la **clase**.

