



Desarrollo de aplicaciones distribuidas con .NET:
Servicios Web en .NET



Grupo ARCOS

Desarrollo de Aplicaciones Distribuidas

Ingeniería Informática

Universidad Carlos III de Madrid

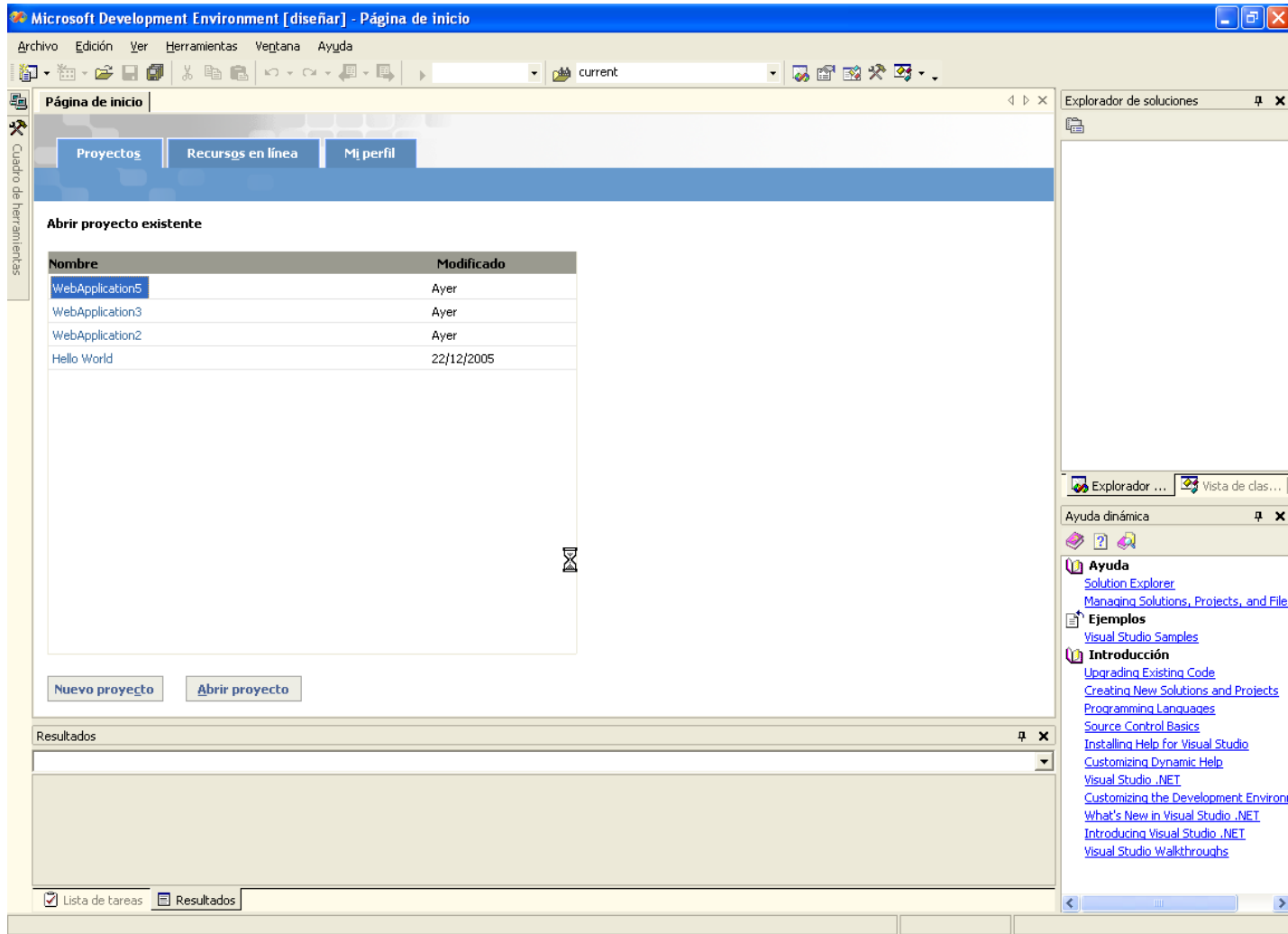
Contenidos

1. **Introducción Visual Studio .NET**
2. **Ejemplo con .NET SDK**
3. **Ejemplo con Visual Studio 2008**

Contenidos

1. **Introducción a Visual Studio .NET**
2. **Ejemplo con .NET SDK**
3. **Ejemplo con Visual Studio 2008**

Introducción a Visual Studio .NET



Unidades en Visual Studio .NET

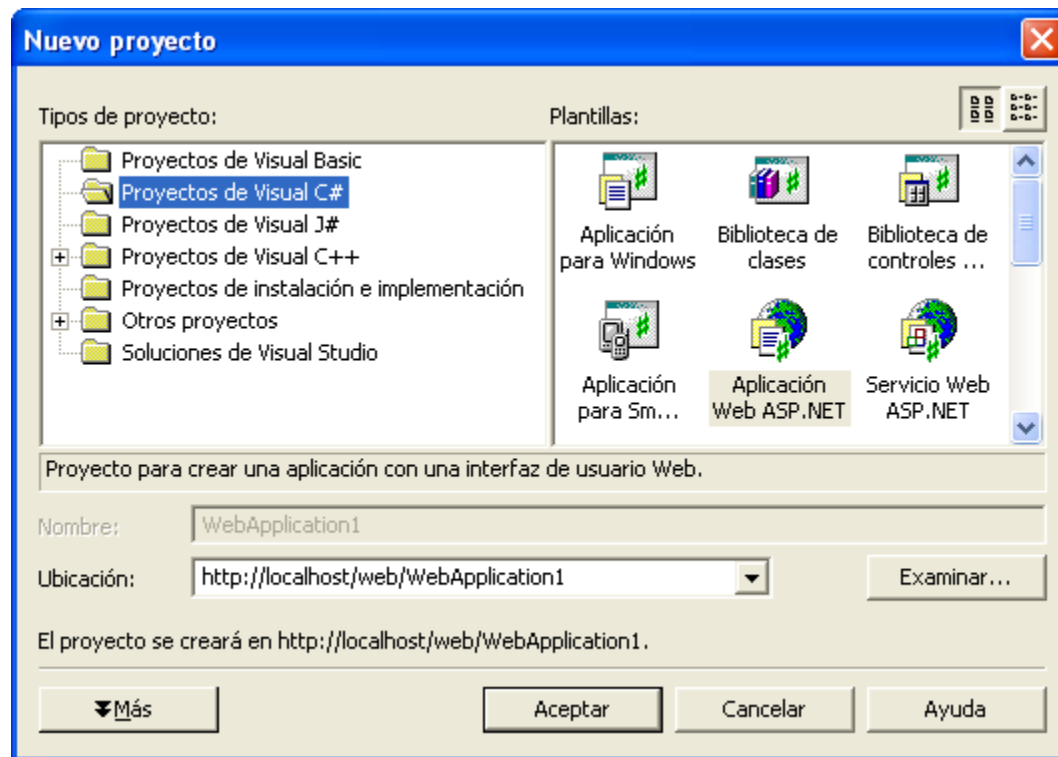
▶ Soluciones

- ▶ Una solución es el punto de inicio para la creación de aplicaciones con Visual Studio
- ▶ Es el contenedor que permite almacenar todas las partes individuales que formarán la aplicación
- ▶ Una solución contiene uno o más proyectos

▶ Proyectos

- ▶ Un proyecto se puede crear utilizando cualquier lenguaje de Visual Studio .NET.
- ▶ Contiene una serie de ficheros
- ▶ El IDE organiza soluciones, proyectos y ficheros de un trabajo de forma jerárquica

Distintos tipos de proyectos



Entorno integrado de desarrollo (IDE)

Cuadro de herramientas

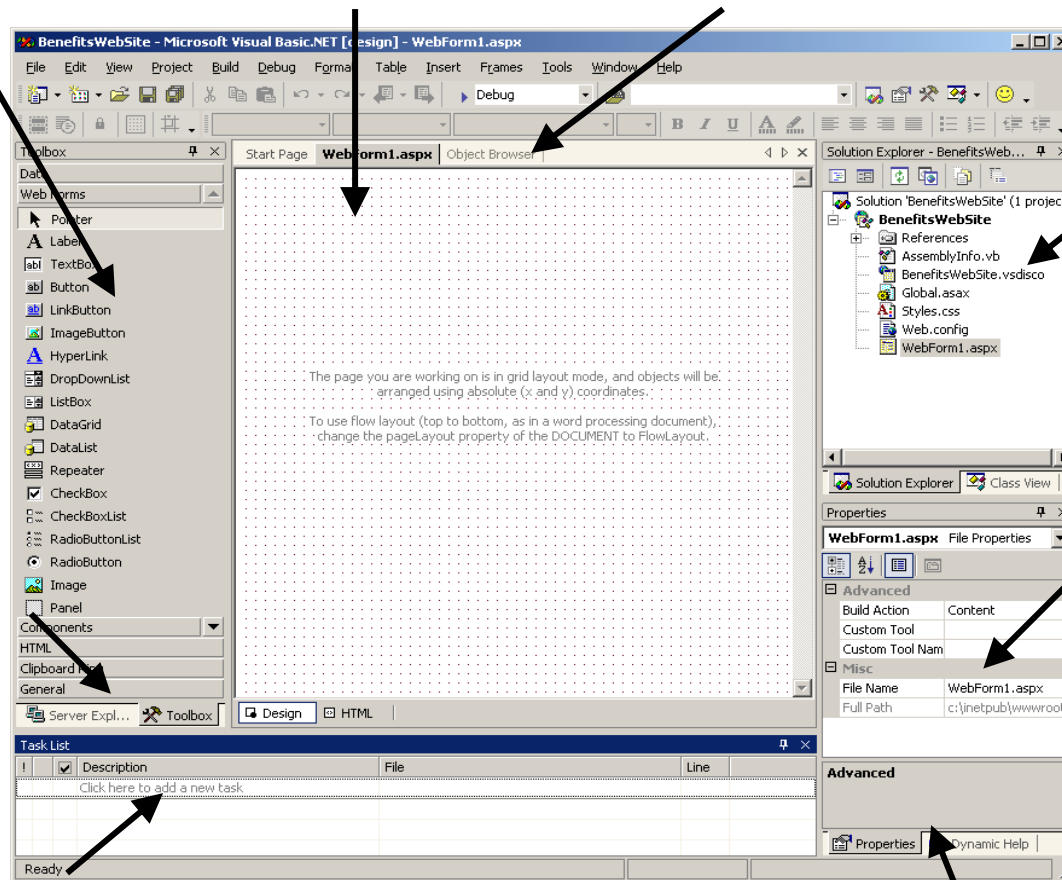
Editor/Navegador

Examinador de objetos

Explorador de soluciones

Explorador de servidores

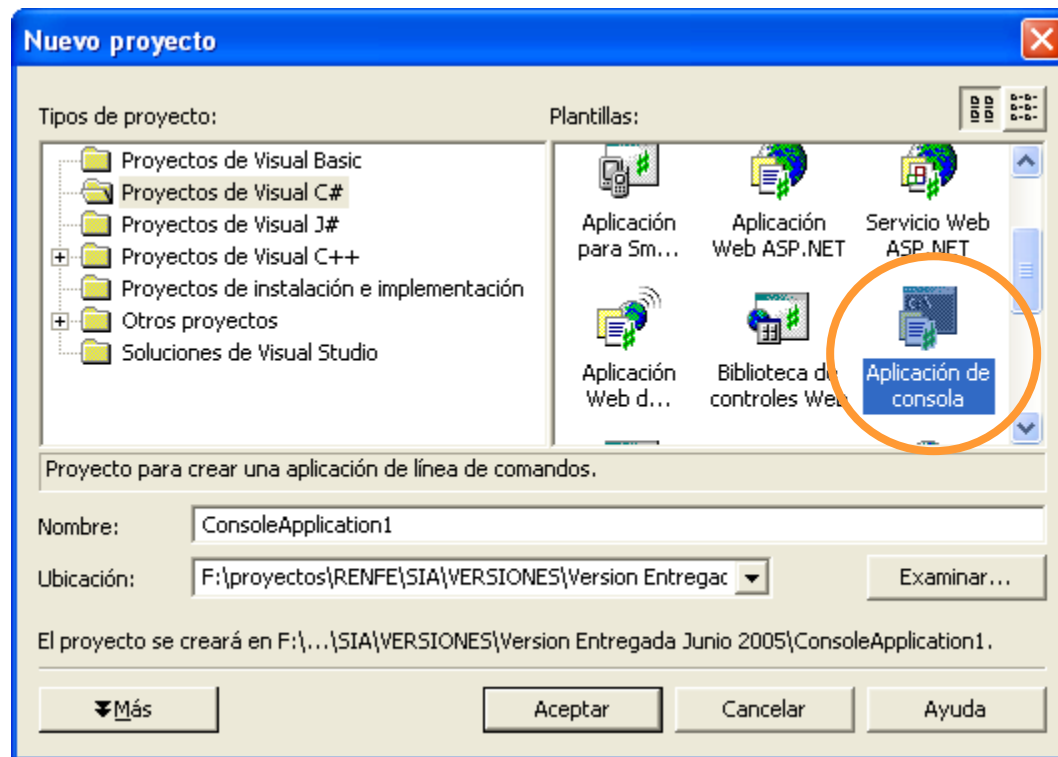
Propiedades



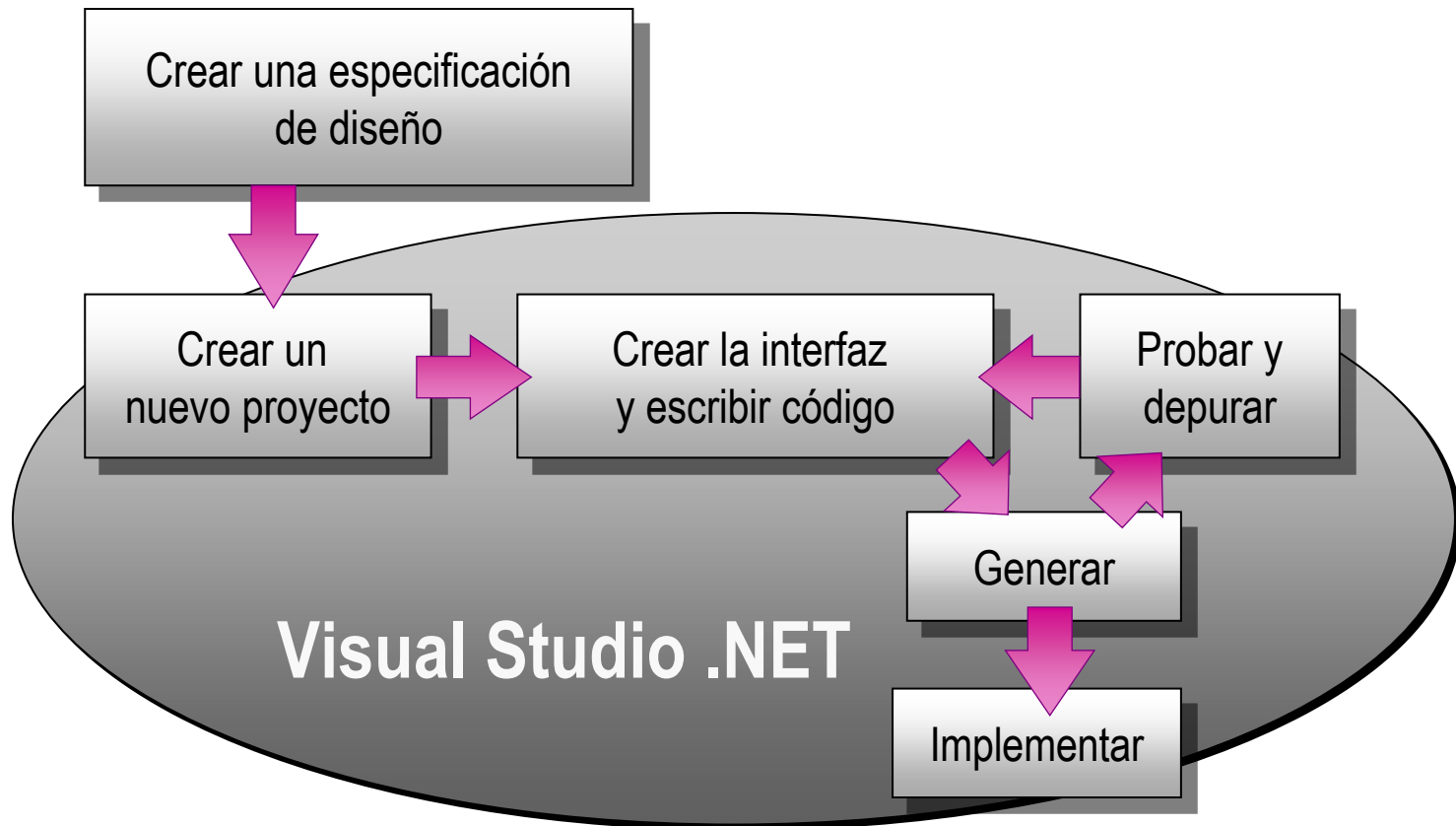
Lista de tareas

Ayuda dinámica

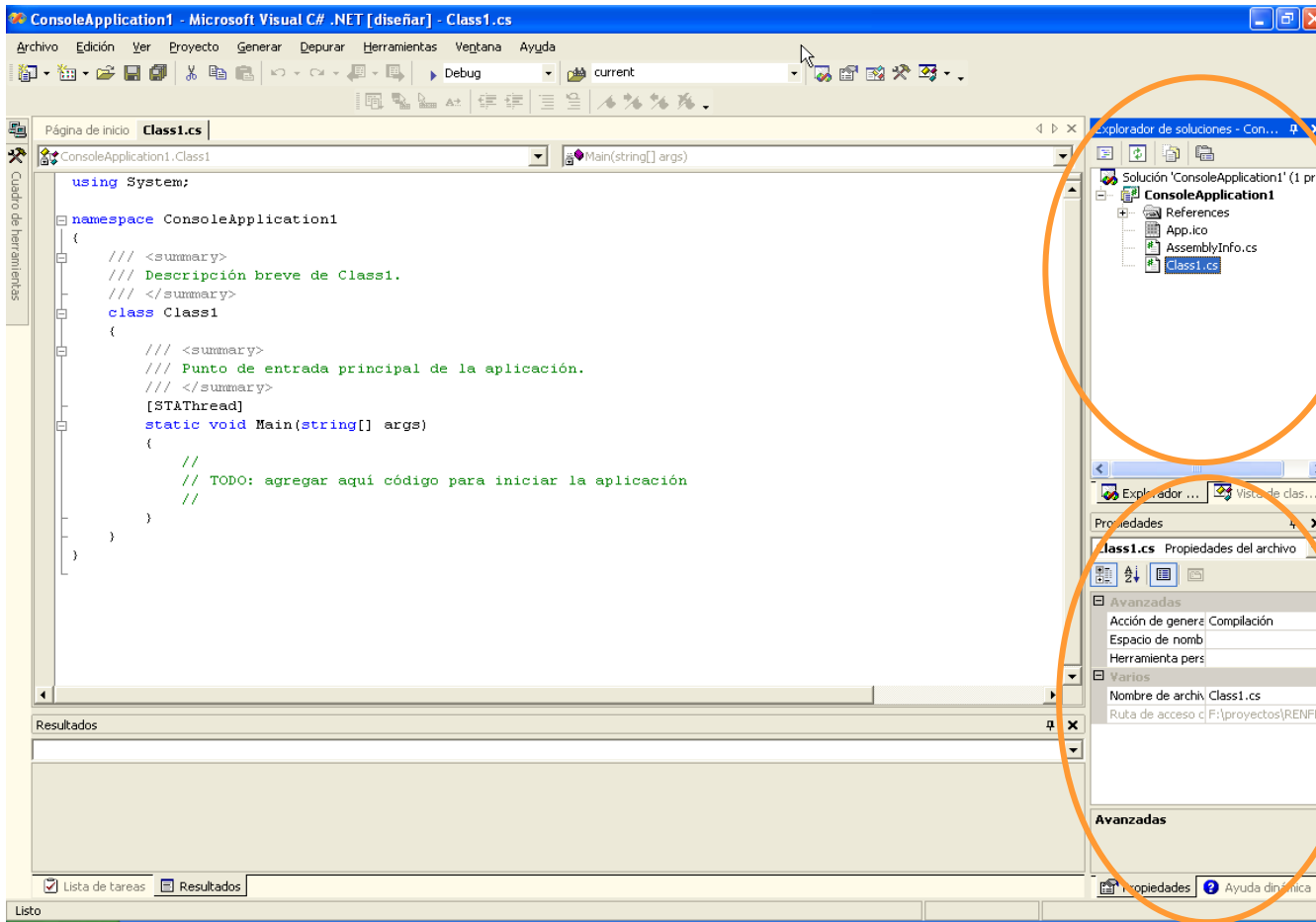
Ejemplo: Aplicación de consola



El proceso de desarrollo



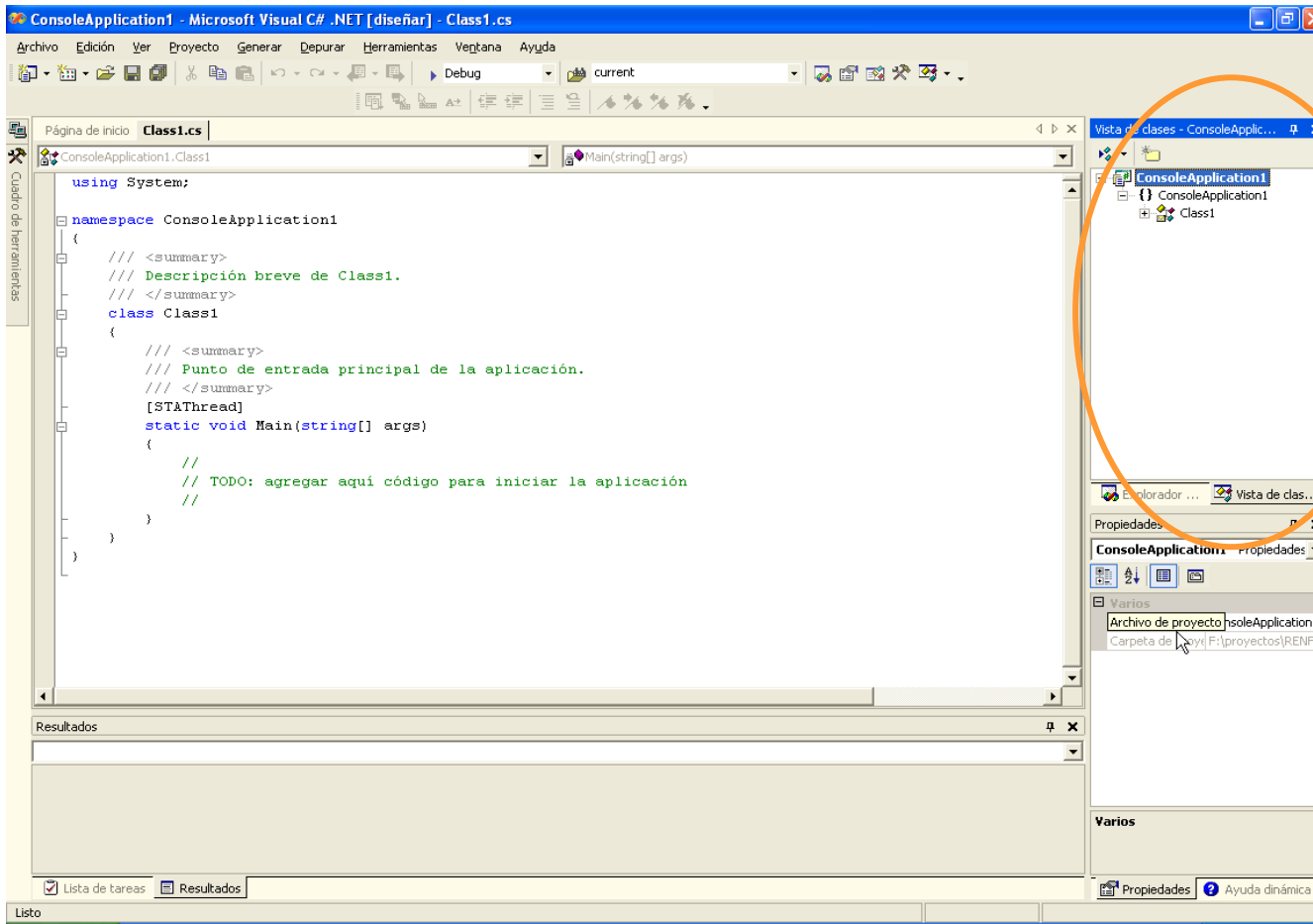
Exploradores



Explorador de Soluciones

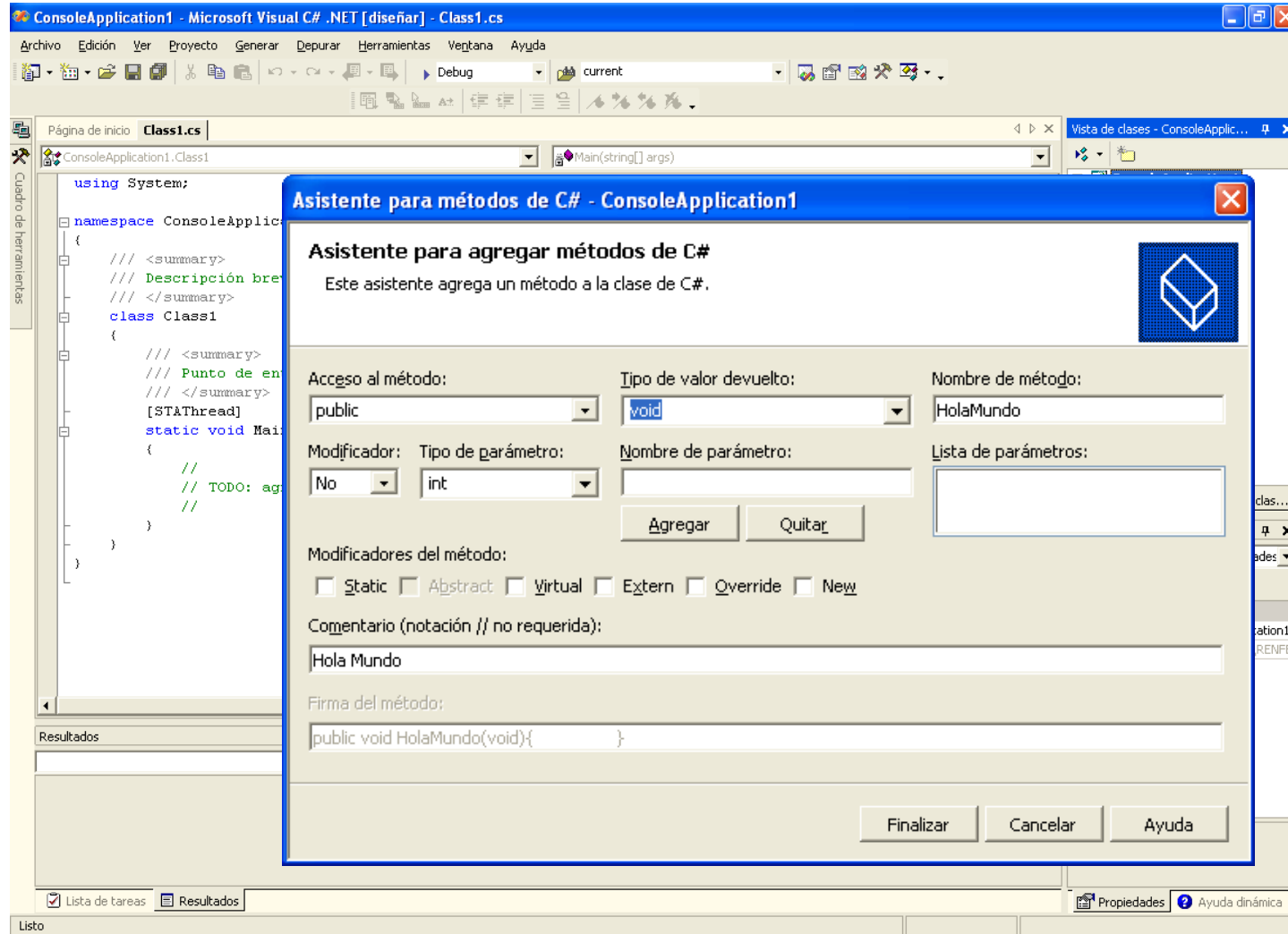
Explorador de Propiedades

Vista de clases

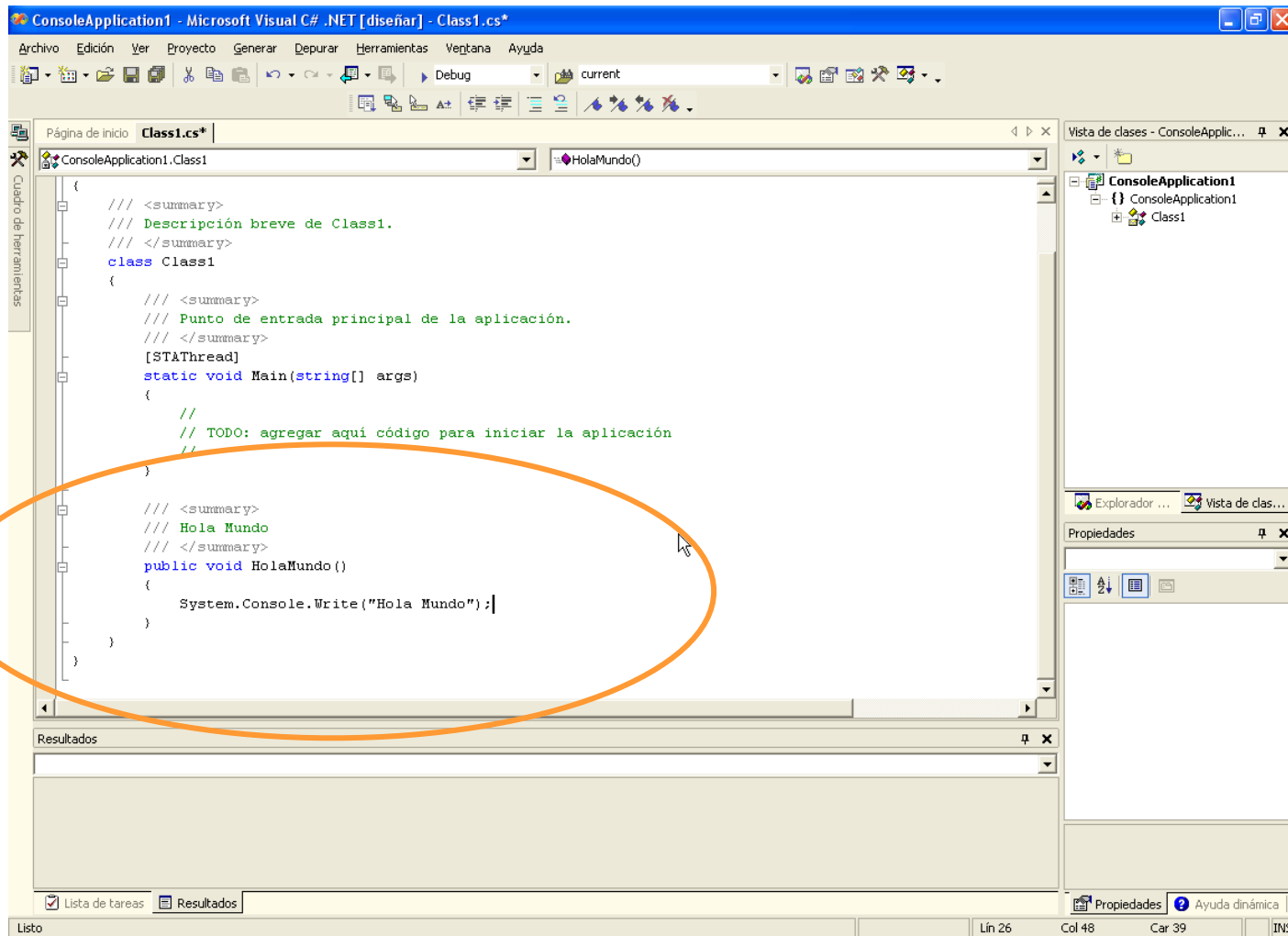


Vista de
clases

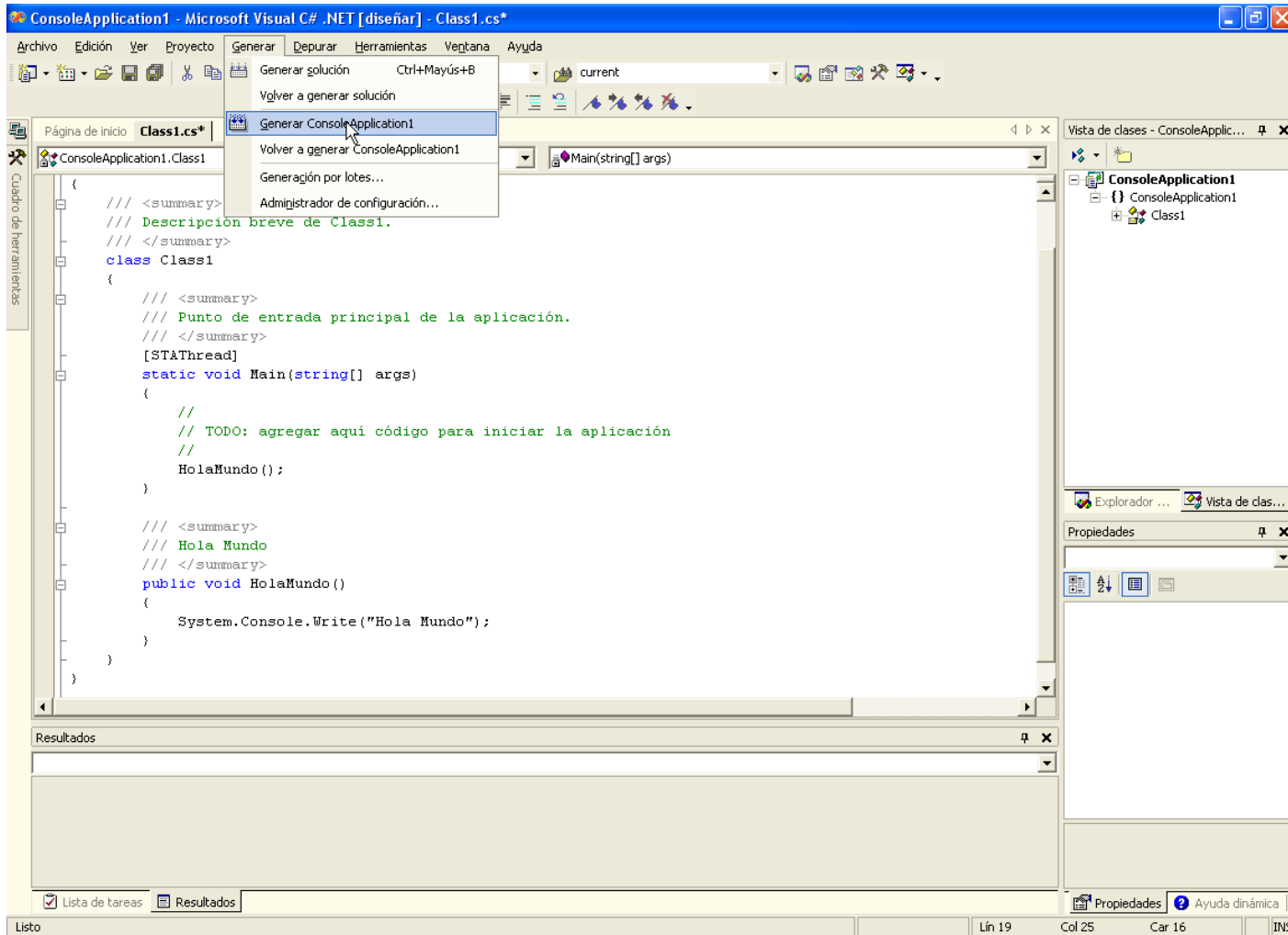
Creación de un método nuevo



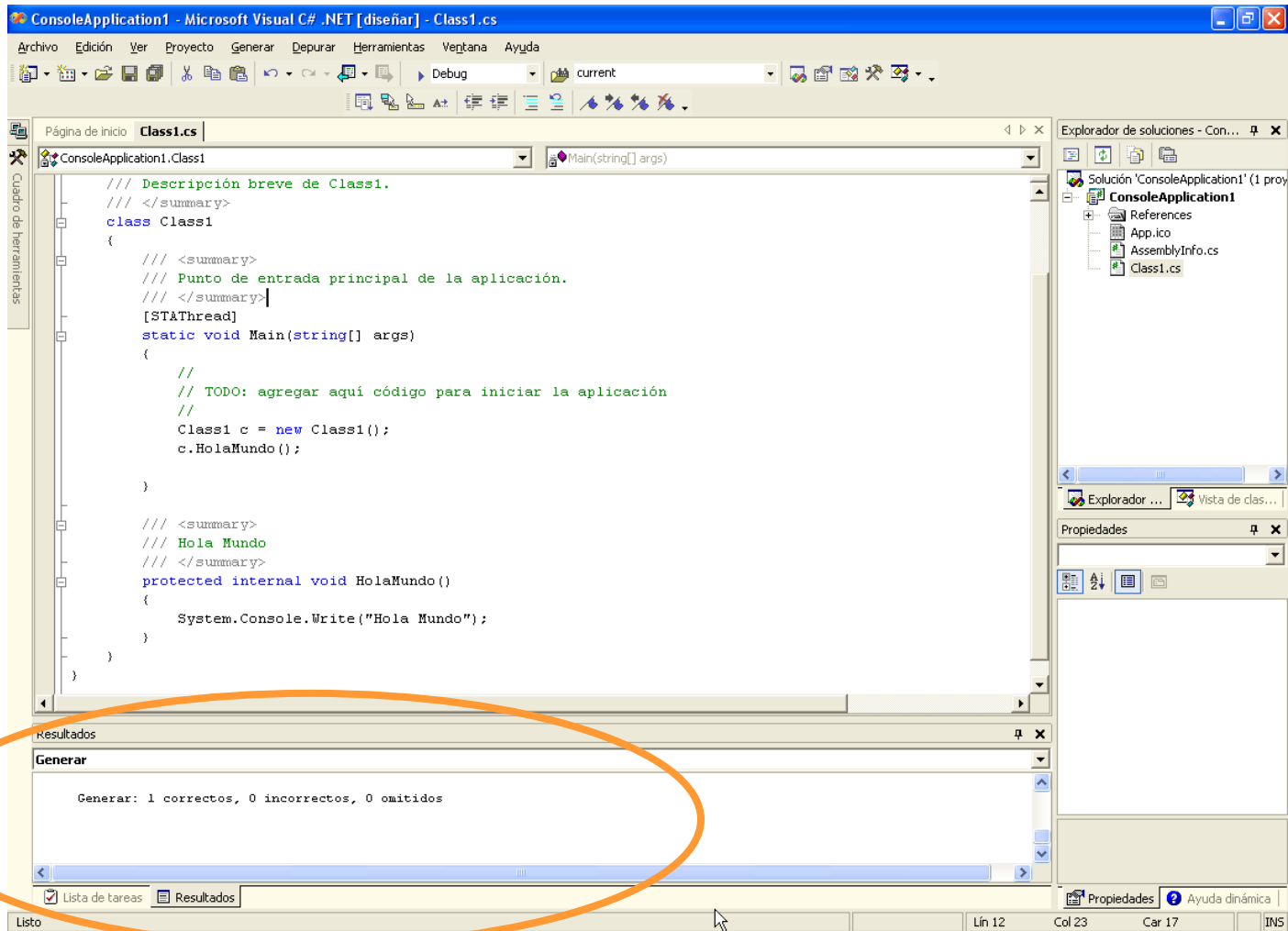
Inserción de código



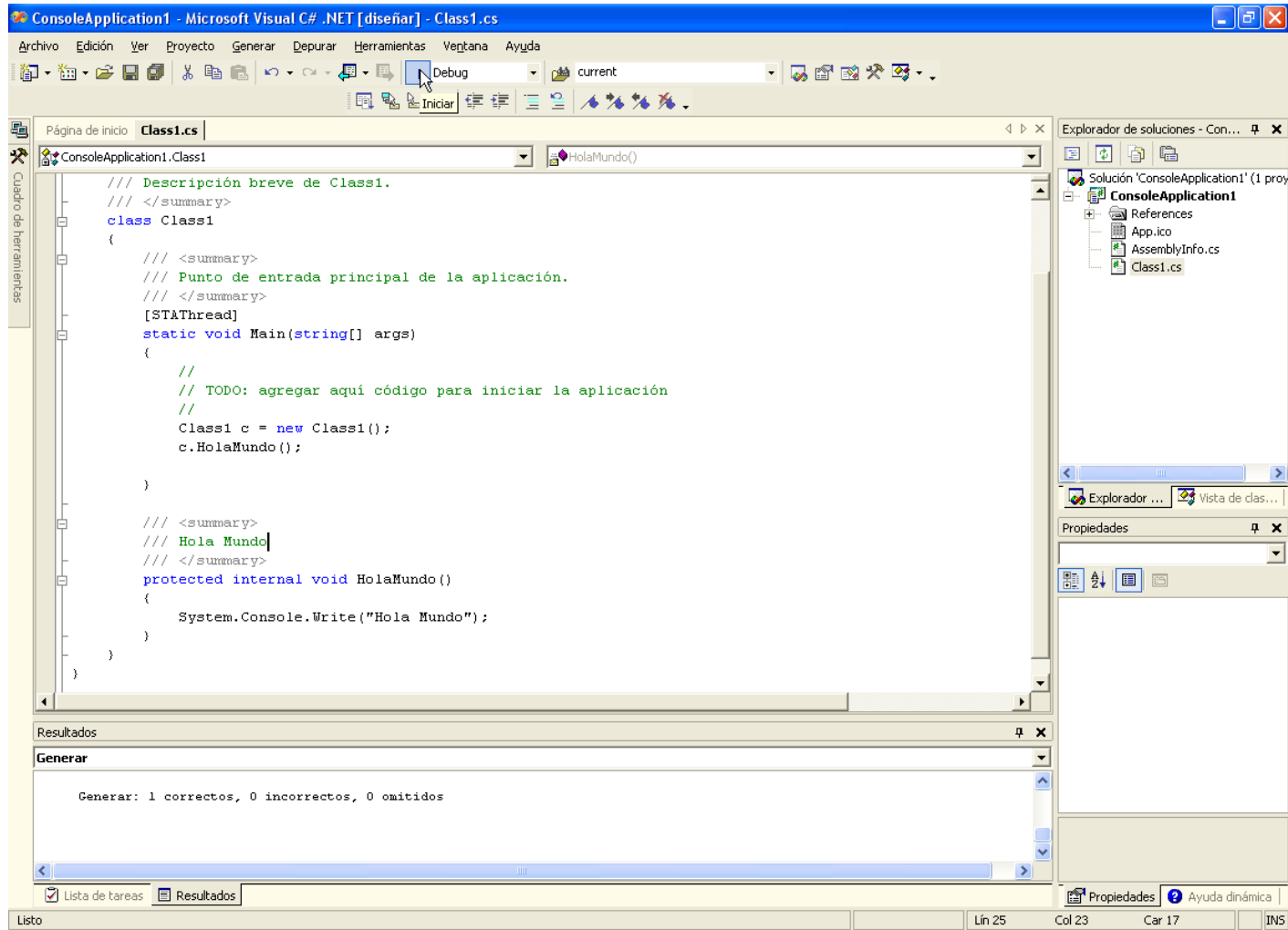
Compilación



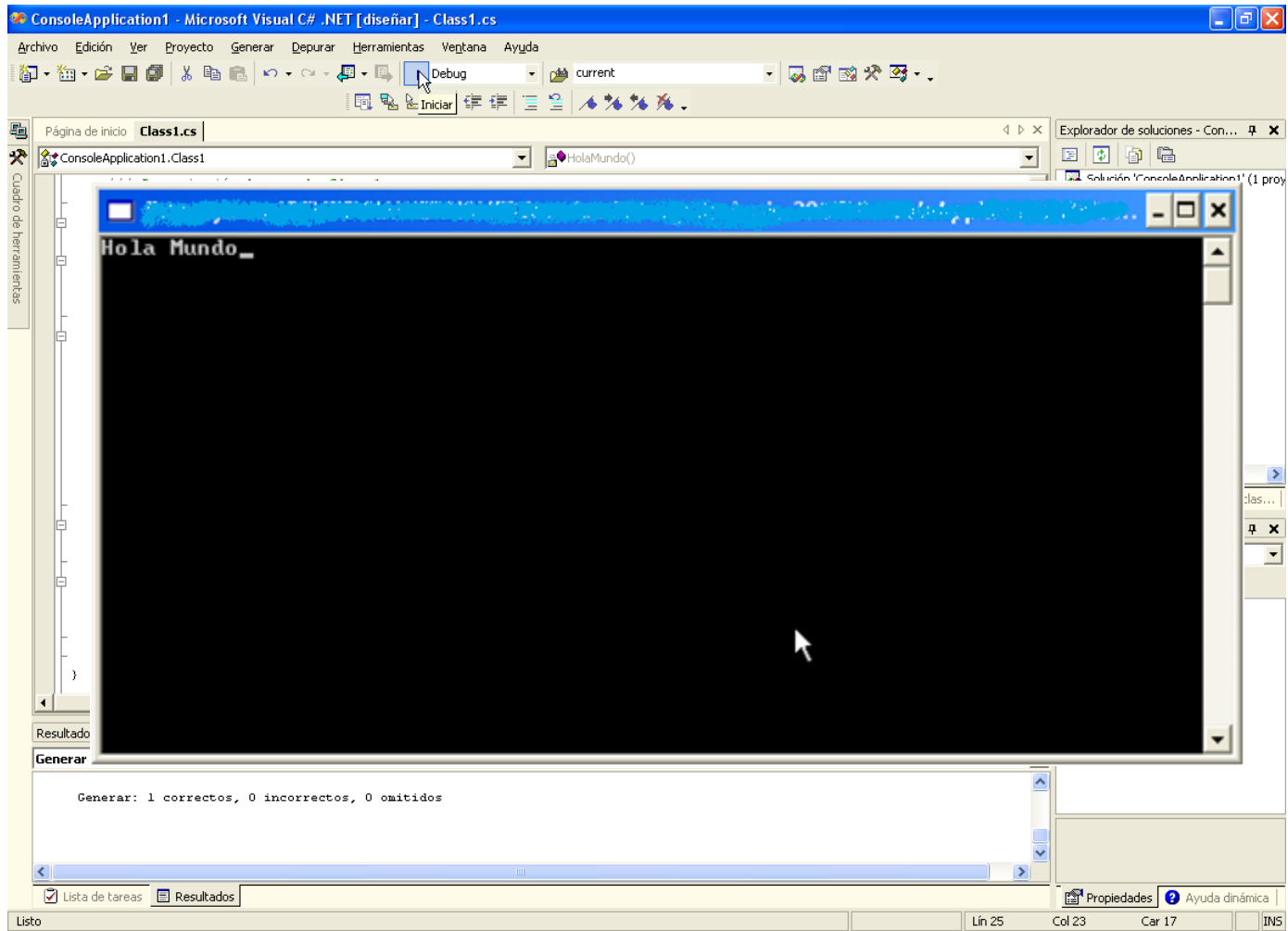
Resultado de la compilación



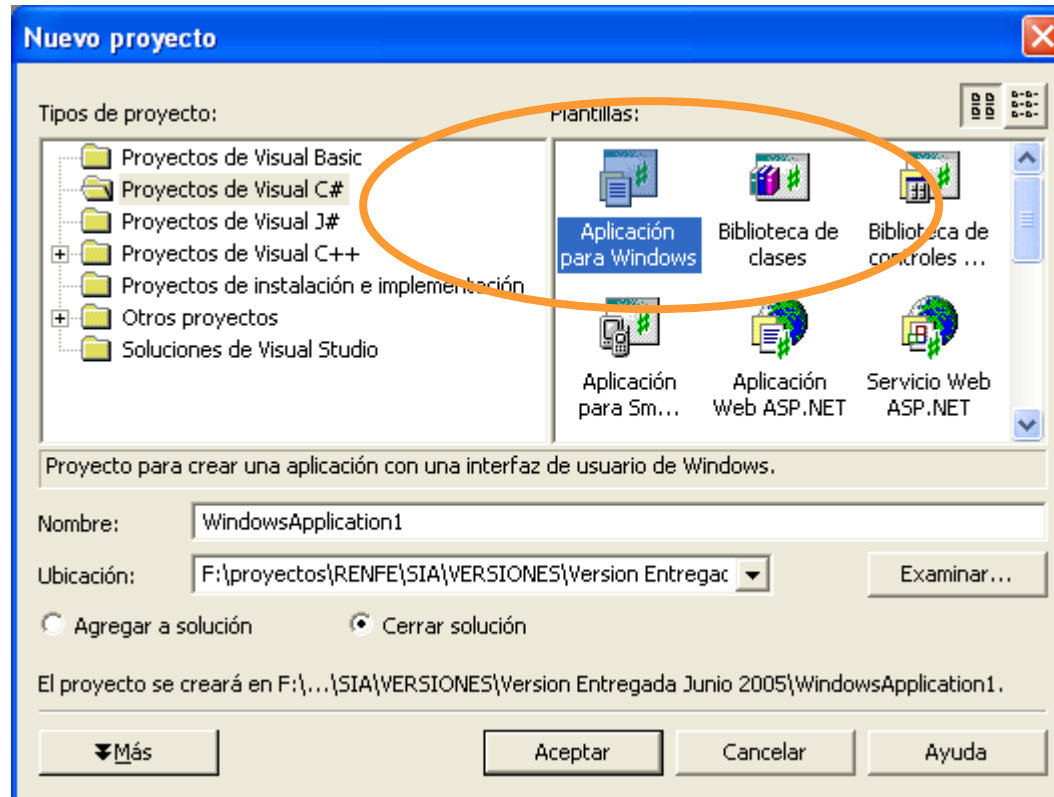
Depuración



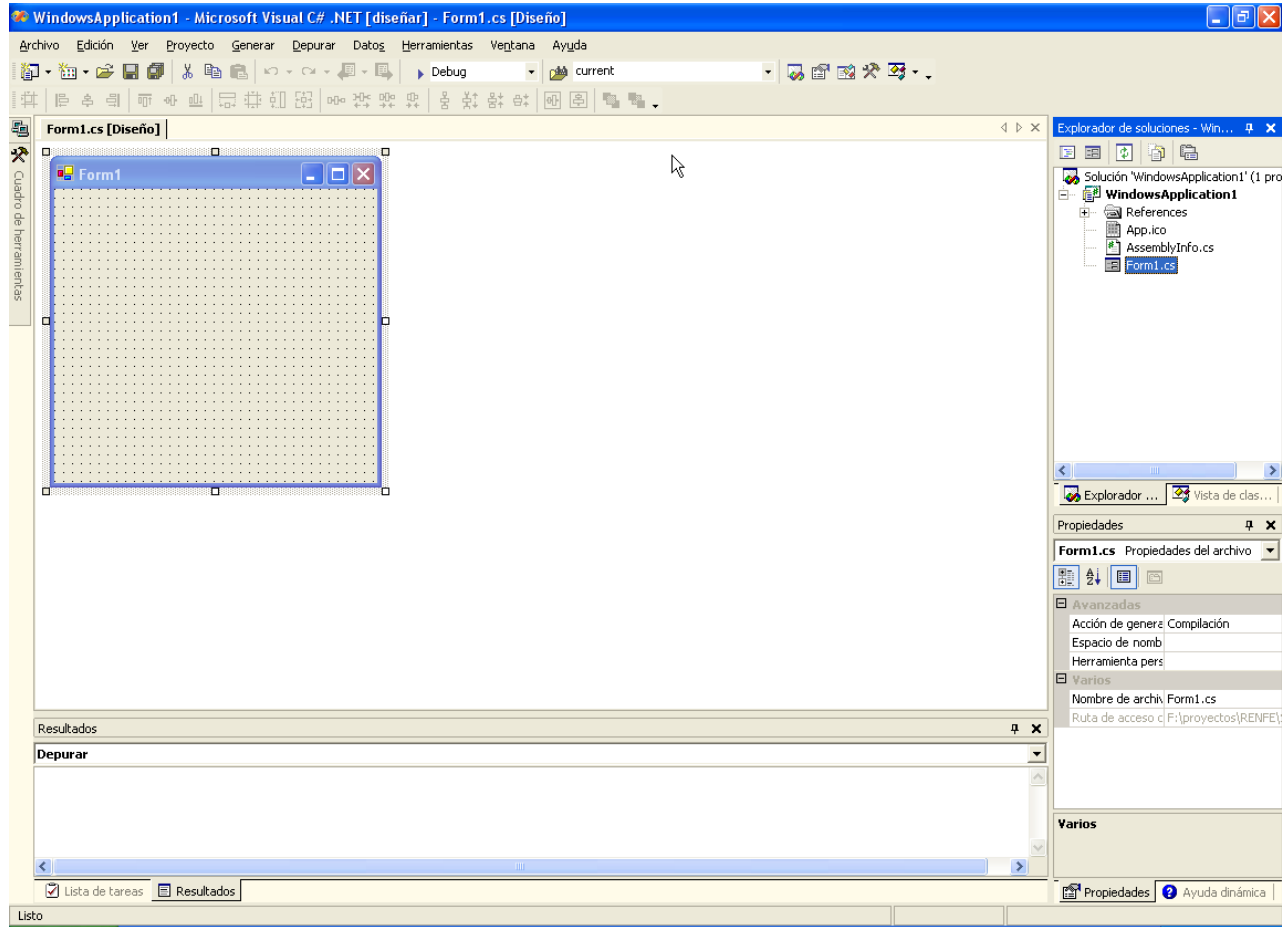
Resultado de la ejecución



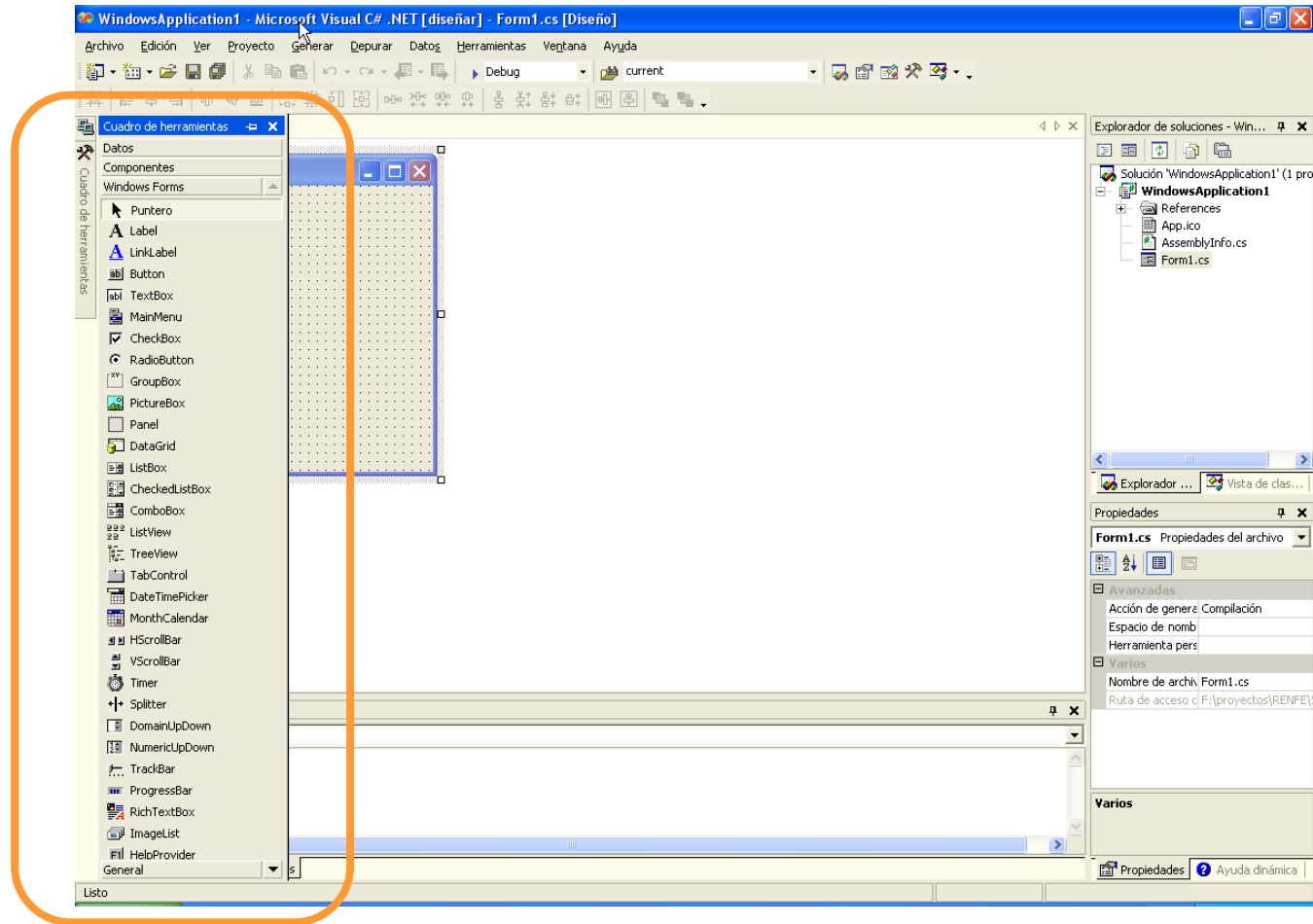
Ejemplo: Aplicación gráfica



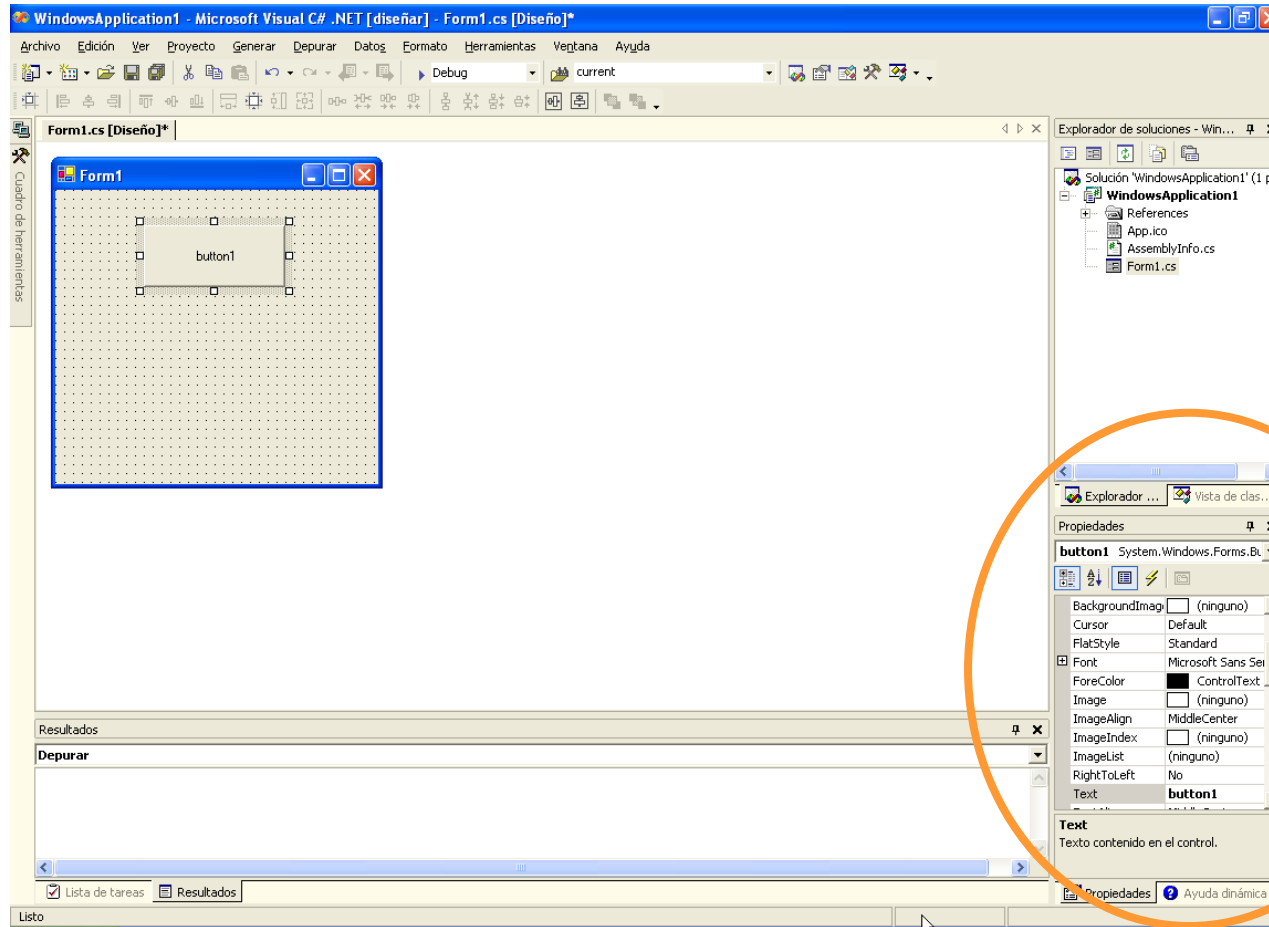
Creación de la interfaz



Cuadro de herramientas

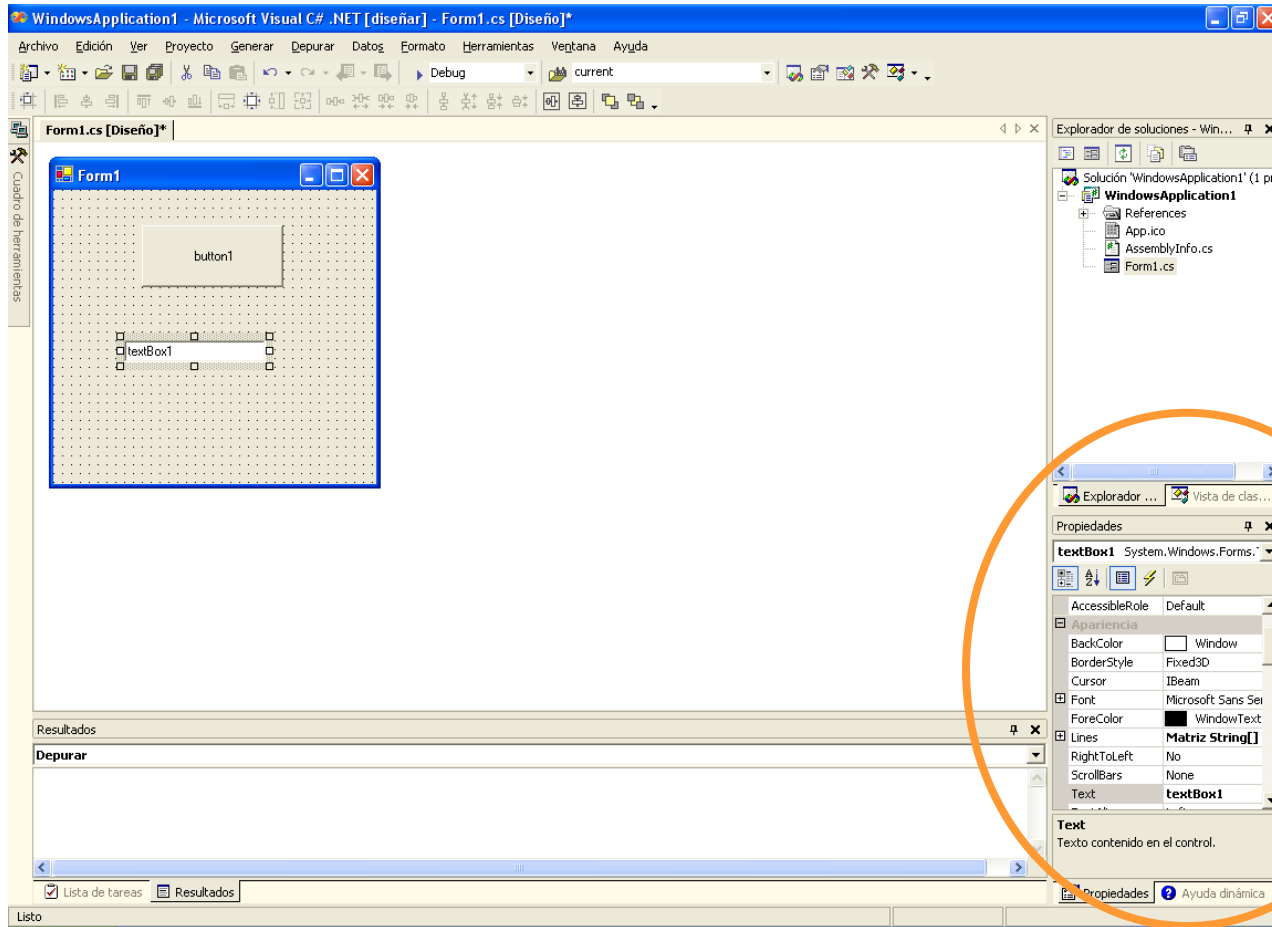


Inserción de un botón



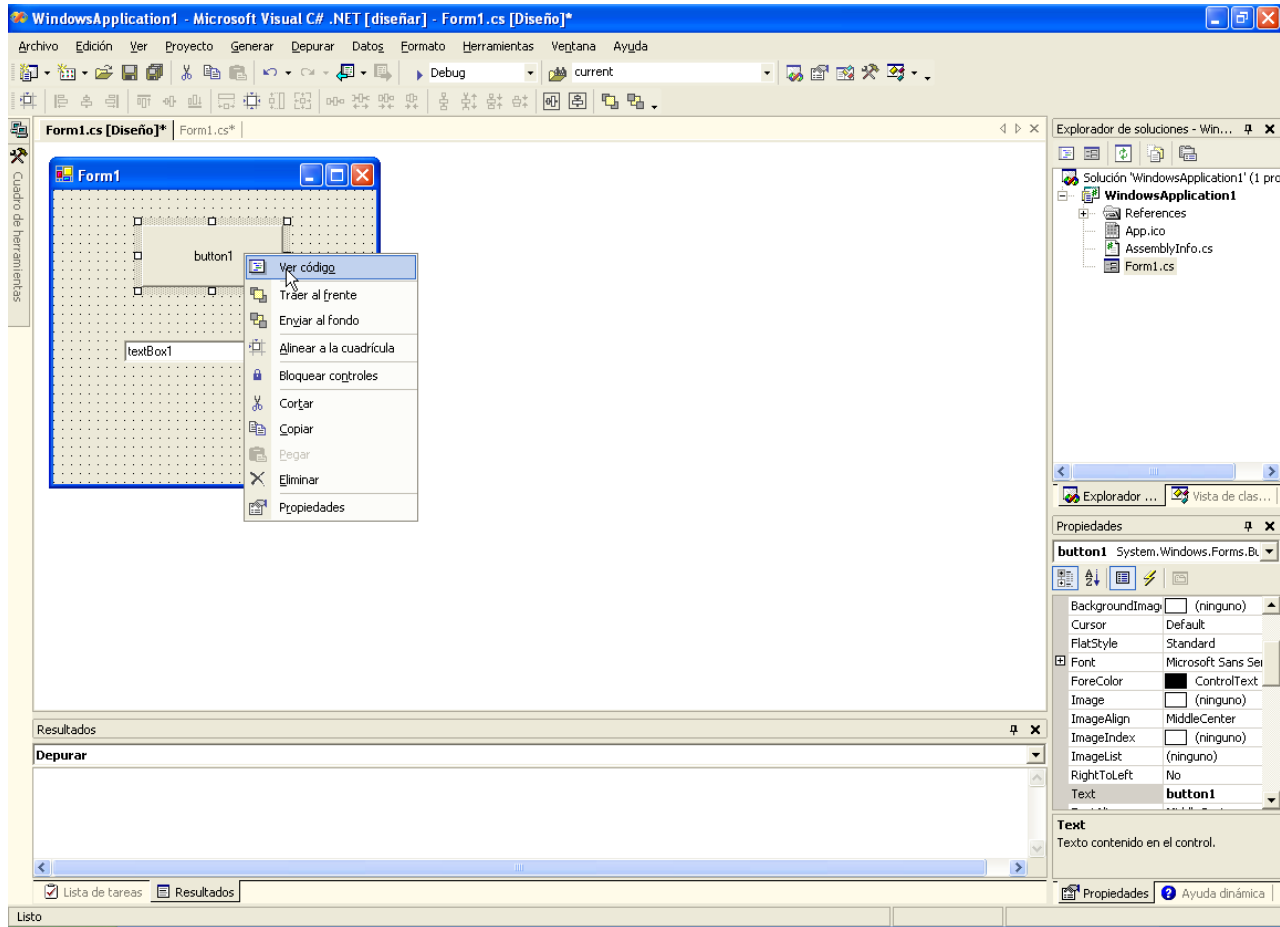
Propiedades
del botón

Cuadro de texto

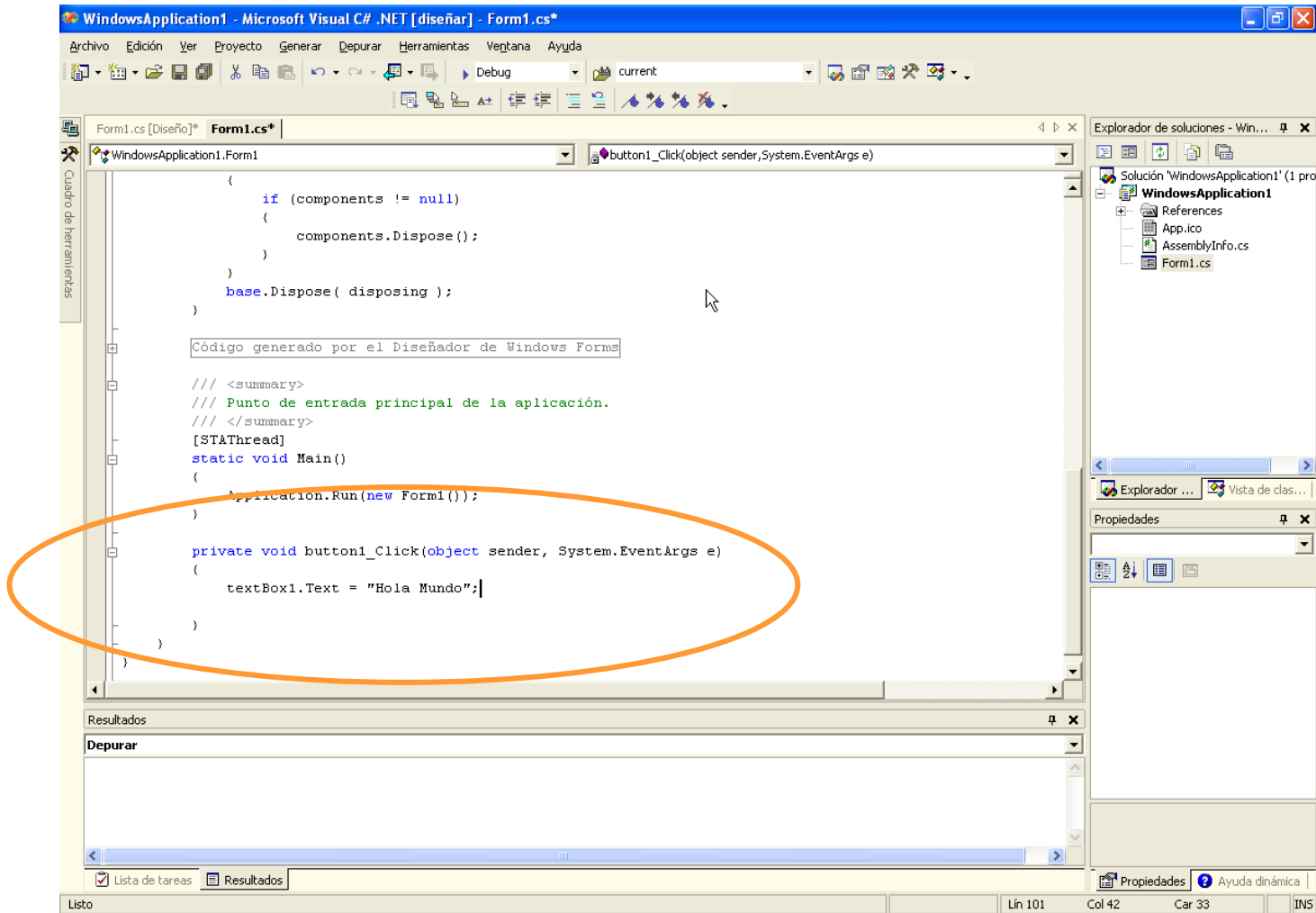


Propiedades
del cuadro

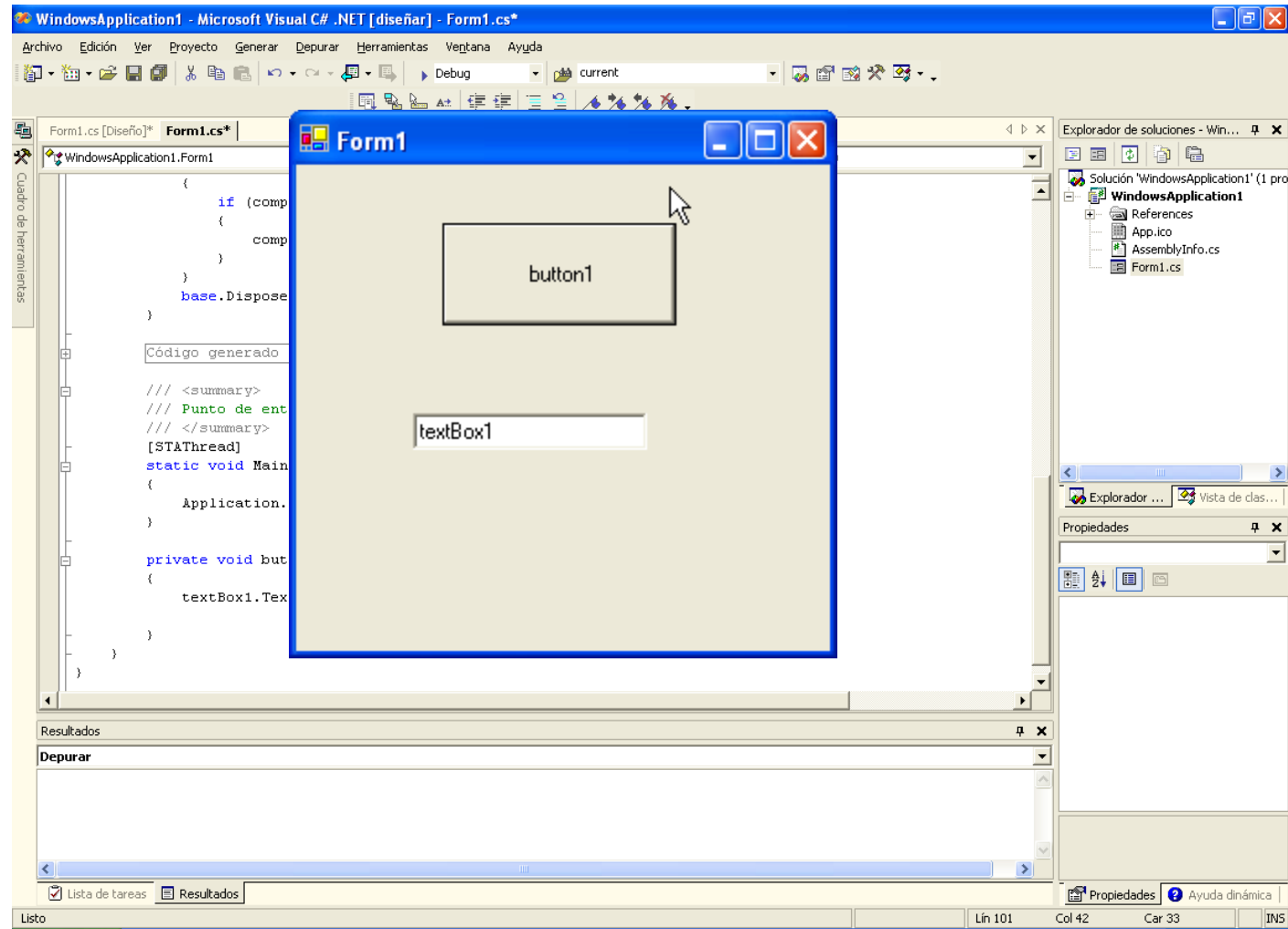
Asignación del código al evento



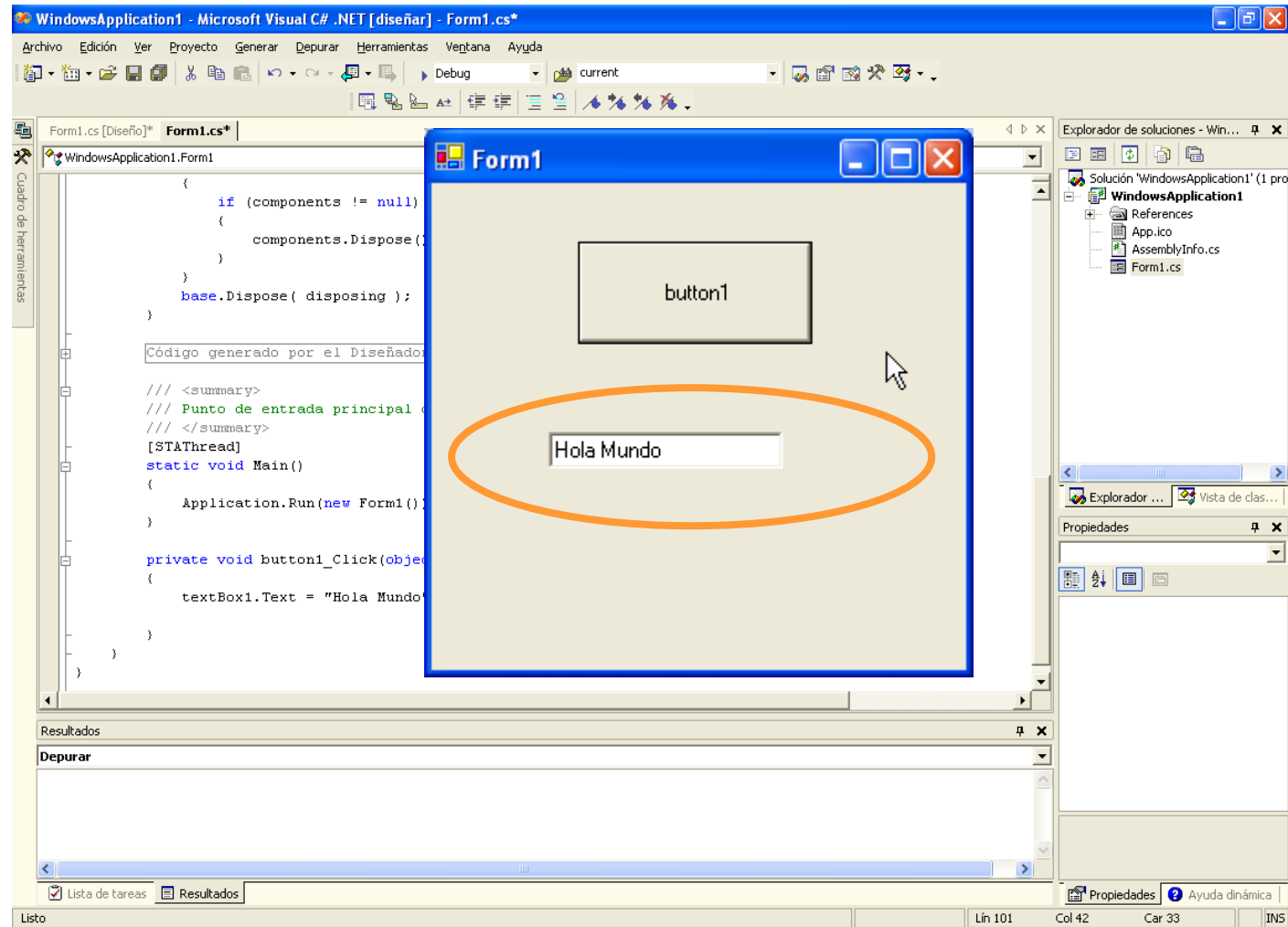
Programación del evento



Ejecución



Ejecución



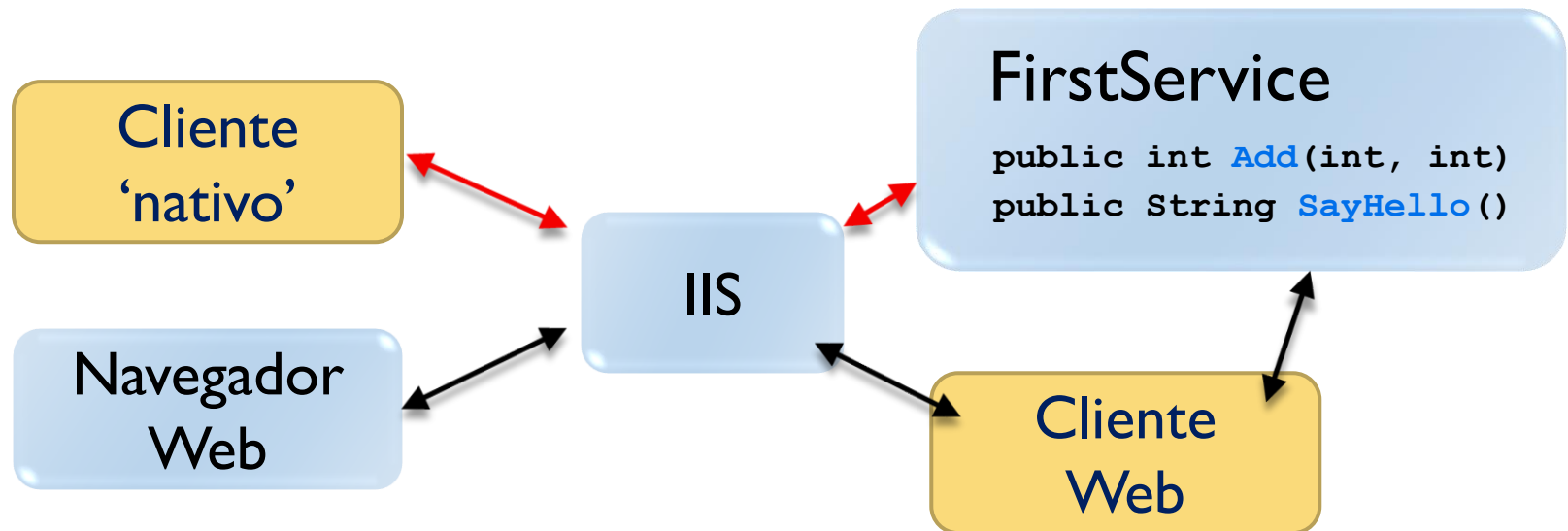
Contenidos

1. Introducción Visual Studio .NET
2. Ejemplo con .NET SDK
3. Ejemplo con Visual Studio 2008



Servicio Web 'hola mundo'

- ▶ Servicio Web **FirstService** con dos métodos, invocado por un **cliente Web** o por un **cliente nativo**



Servicio Web

1. **Codificación del servicio**
2. **Publicación del servicio**
3. **Prueba de que el servicio está activo**

1.- Codificación del servicio Web

▶ Edición:

- ▶ Escritura del servicio web en C
- ▶ Se guardará en el archivo `c:\temp\FirstService.asmx`

FirstService.asmx

```
<%@ WebService language="C#" class="FirstService" %>

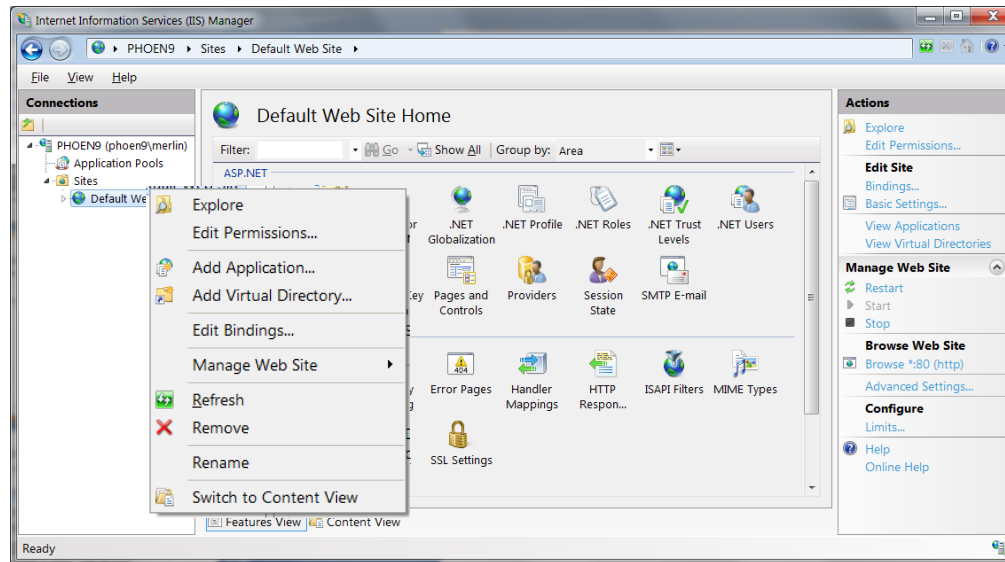
using System;
using System.Web.Services;
using System.Xml.Serialization;

[WebService (Namespace="http://localhost/MyWebServices/")]
public class FirstService : WebService
{
    [WebMethod]
    public int Add(int a, int b)
    {
        return a + b;
    }

    [WebMethod]
    public String SayHello()
    {
        return "Hello World";
    }
}
```

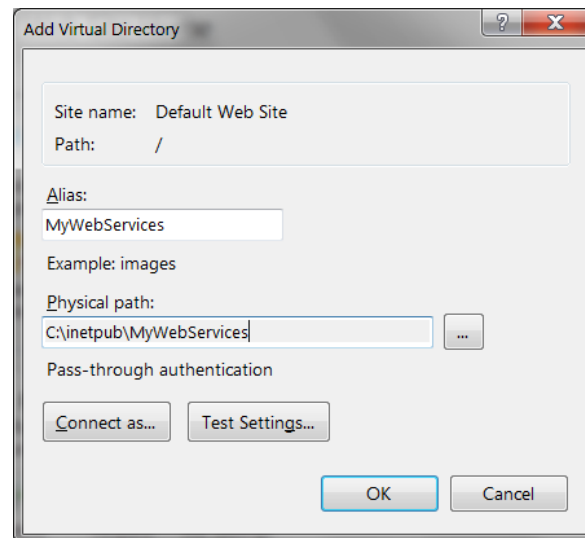
2.- Publicación del servicio Web (w7)

- ▶ Abrir el gestor de IIS:
 - ▶ Windows+R
 - ▶ Inetmgr
- ▶ Abrir el asistente de directorio virtual:
 - ▶ Expandir hasta ver 'Default web site'
 - ▶ Botón derecho y seleccionar 'Add virtual directory...'



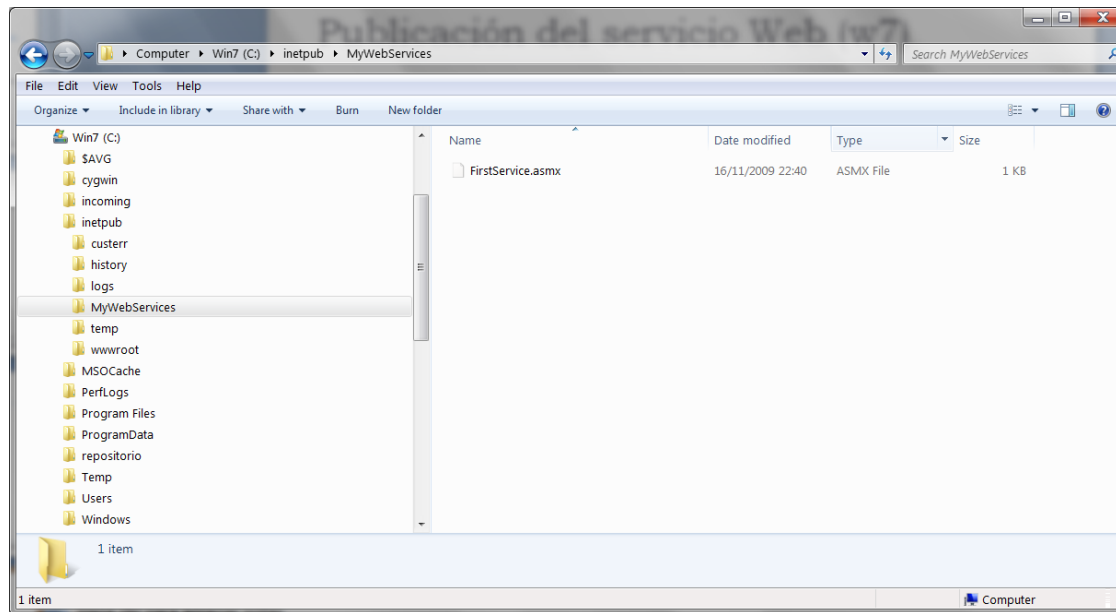
2.- Publicación del servicio Web (w7)

- ▶ Configurar un nuevo directorio virtual:
 - ▶ Indicar el alias (ej.: MyWebServices)
 - ▶ Indicar la ruta real completa asociada (ej.: C:\inetpub\MyWebServices)
 - ▶ Confirmar con ok



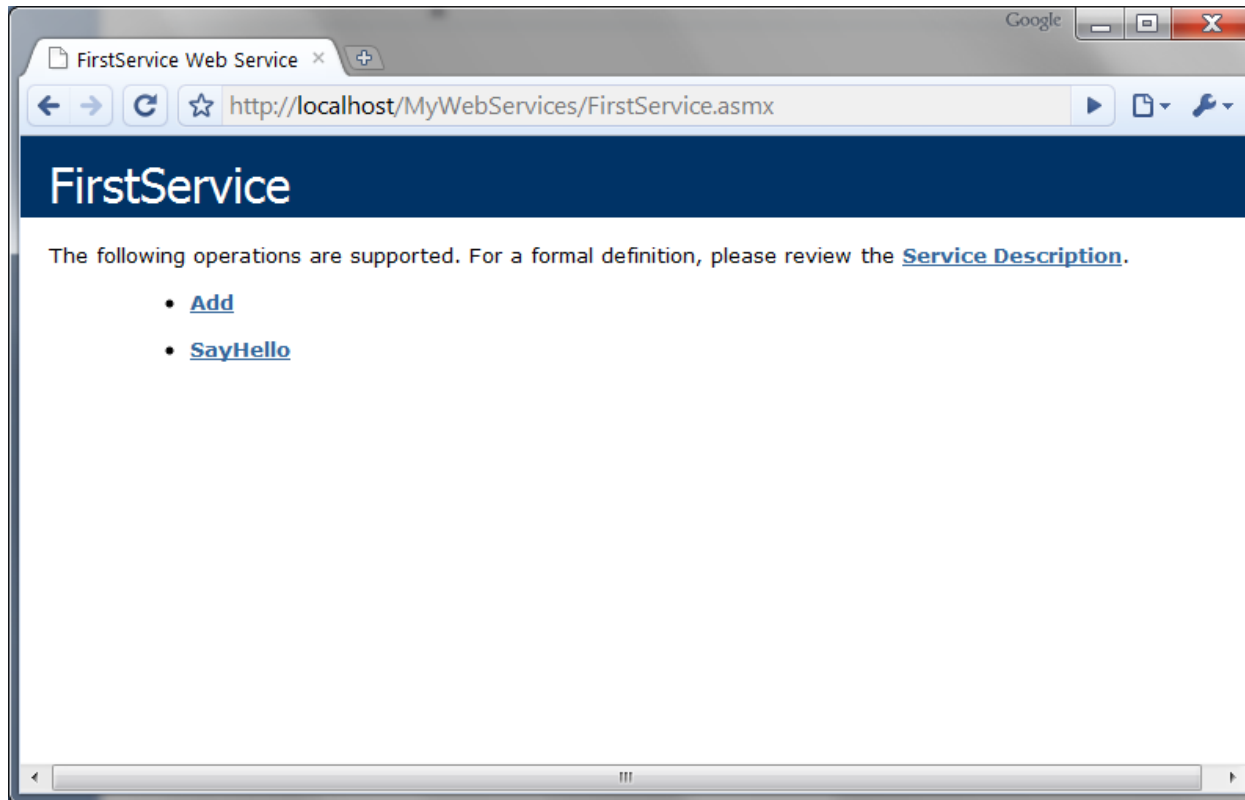
2.- Publicación del servicio Web (w7)

- ▶ Publicar el servicio:
 - ▶ Copiar a C:\inetpub\MyWebServices el archivo c:\temp\FirstService.asmx
 - ▶ Confirmar como administrador
 - ▶ Es posible usar un directorio virtual por servicio web o un único directorio virtual para todos los servicios web



3.- Probar que el servicio está activo

- ▶ Abrir la página en navegador web:
 - ▶ <http://localhost/MyWebServices/FirstService.asmx>



3.- Probar que el servicio está activo

- ▶ Probar cada método:
 - ▶ Sayhello
 - ▶ Add

The screenshot shows a web browser window with the address bar containing `http://localhost/MyWebServices/FirstService.asmx?op=SayHello`. The page title is "FirstService". Below the title, there is a link "Click [here](#) for a complete list of operations." The main heading is "SayHello". Underneath, there is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." and an "Invoke" button. Below the "Test" section, there is a "SOAP 1.1" section with the text "The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values." and a code block containing a sample SOAP 1.1 request and response.

```
POST /MyWebServices/FirstService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/MyWebServices/SayHello"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SayHello xmlns="http://localhost/MyWebServices/" />
  </soap:Body>
</soap:Envelope>
```

The screenshot shows a web browser window with the address bar containing `http://localhost/MyWebServices/FirstService.asmx?op=Add`. The page title is "FirstService". Below the title, there is a link "Click [here](#) for a complete list of operations." The main heading is "Add". Underneath, there is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." and an "Invoke" button. Below the "Test" section, there is a "SOAP 1.1" section with the text "The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values." and a code block containing a sample SOAP 1.1 request and response.

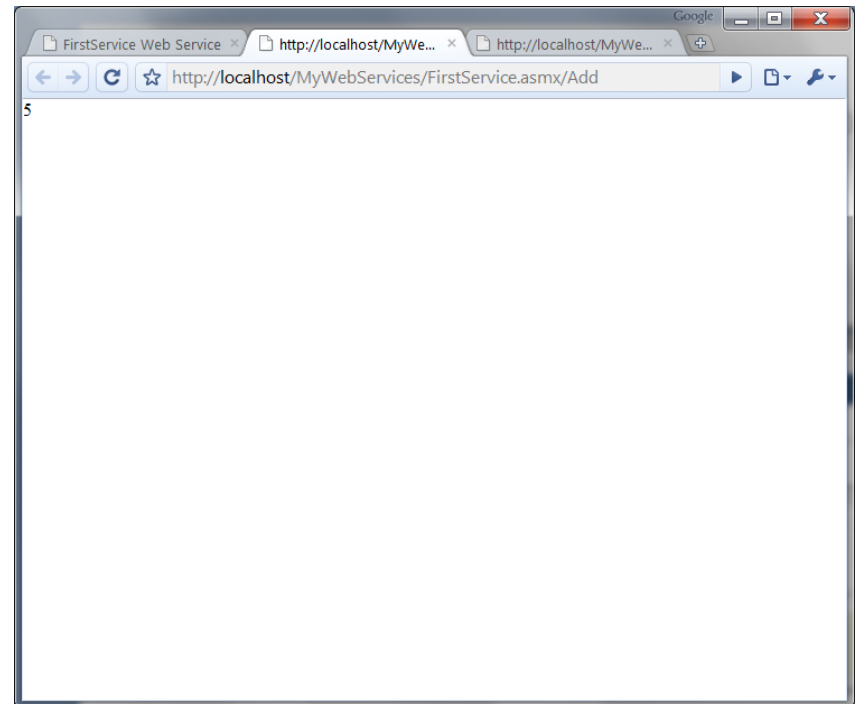
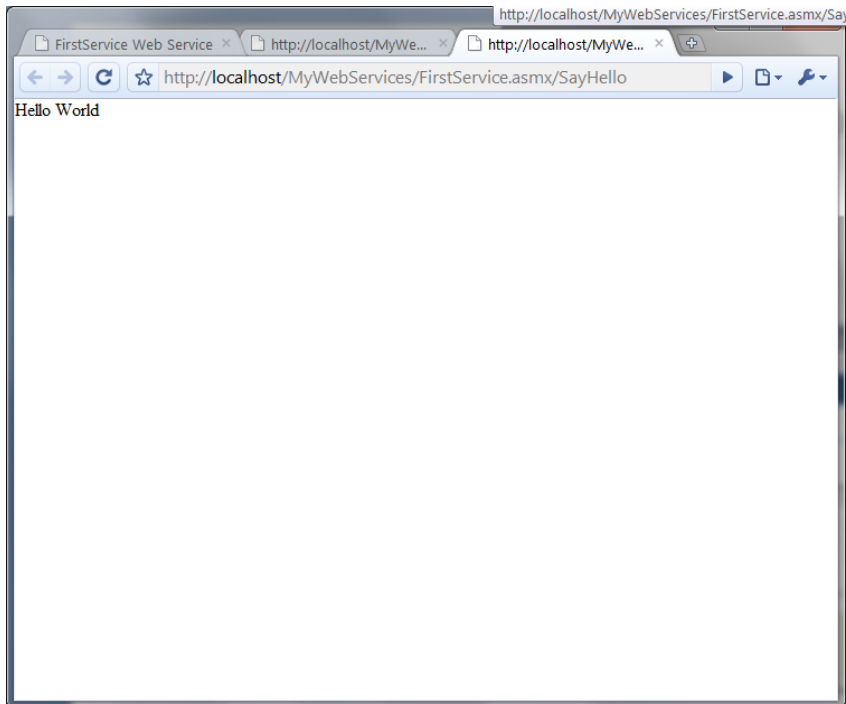
Parameter	Value
a:	<input type="text"/>
b:	<input type="text"/>

```
POST /MyWebServices/FirstService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/MyWebServices/Add"

<?xml version="1.0" encoding="utf-8"?>
```

3.- Probar que el servicio está activo

- ▶ Probar cada método:
 - ▶ `SayHello`
 - ▶ `Add`



Consumidor del servicio Web

Aplicación Web

1. Creación del proxy
2. Creación del cliente Web
3. Publicación del cliente Web
4. Ejecución de la aplicación Web

1.- Creación del proxy

▶ Preprocesado:

```
C:\temp> cmd.exe /V:ON /E:ON /K "C:\Program Files\Microsoft SDKs\ →  
→ windows\v6.1\bin\setenv.cmd" /Release  
C:\temp> WSDL http://localhost/MyWebServices/FirstService.asmx?WSDL
```

▶ Compilación:

```
C:\Temp\> csc /t:library FirstService.cs
```

▶ Despliegue:

- ▶ Copiar de `c:\temp\FirstService.dll`
a `C:\inetpub\MywebServices\bin`

2.- Creación del cliente Web

- ▶ Edición:

- ▶ Escritura del consumidor web en ASP.NET
- ▶ Se guardará en el archivo `c:\temp\WebApp.aspx`

WebApp.aspx (1 / 3)

```
<%@ Page
Language="C#"
CompilerOptions=' /R:"C:\inetpub\MyWebServices\bin\FirstService.dll"'
%>

<script runat="server">
void runSrvce_Click(Object sender, EventArgs e)
{
    FirstService mySvc = new FirstService();
    Label1.Text = mySvc.SayHello();
    Label2.Text = mySvc.Add(Int32.Parse(txtNum1.Text),
                          Int32.Parse(txtNum2.Text)).ToString();
}
</script>
<html>
<head>
</head>
<body>
```

WebApp.aspx (2 / 3)

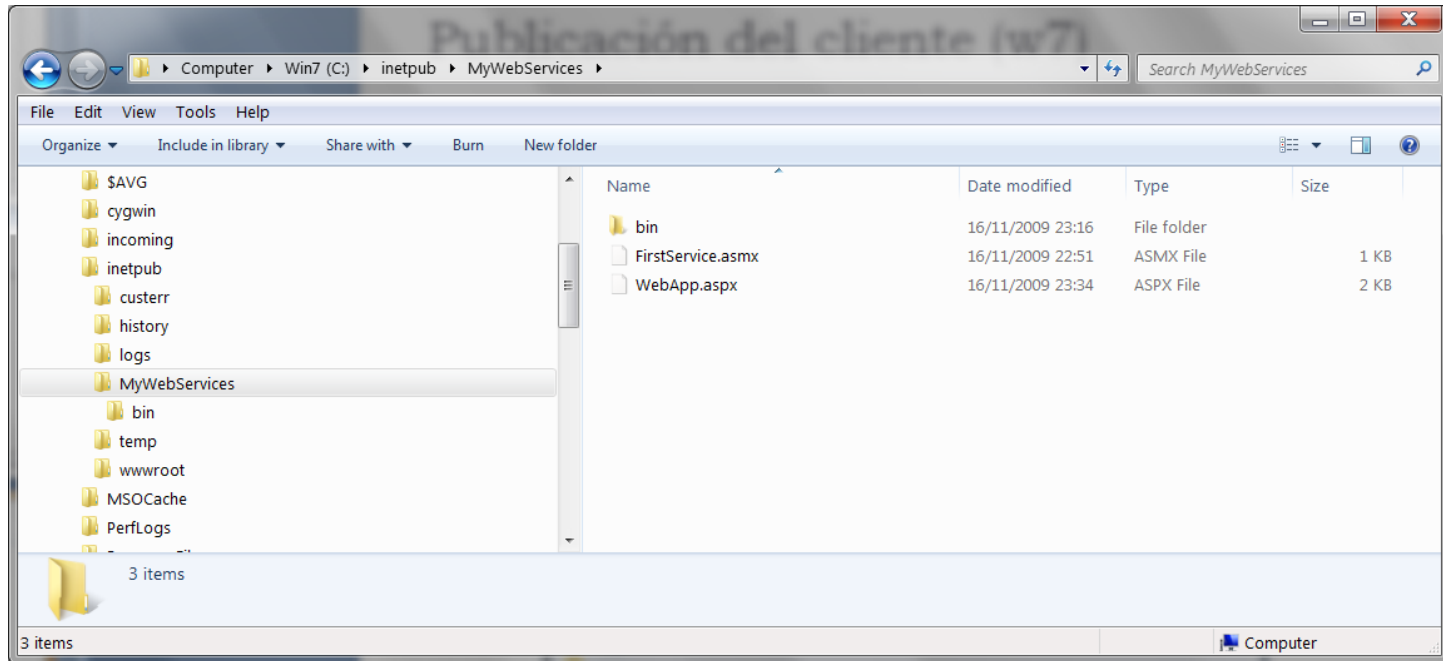
```
<form runat="server">
  <p>
    <em>First Number to Add </em>:
    <asp:TextBox id="txtNum1" runat="server"
      Width="43px">4</asp:TextBox>
  </p>
  <p>
    <em>Second Number To Add </em>:
    <asp:TextBox id="txtNum2" runat="server"
      Width="44px">5</asp:TextBox>
  </p>
  <p>
    <strong><u>Web Service Result -</u></strong>
  </p>
```

WebApp.aspx (3 / 3)

```
<p>
  <em>Hello world Service</em> :
  <asp:Label id="Label1" runat="server"
    Font-Underline="True">Label</asp:Label>
</p>
<p>
  <em>Add Service</em> :
  & <asp:Label id="Label2" runat="server"
    Font-Underline="True">Label</asp:Label>
</p>
<p align="left">
  <asp:Button id="runSrvice" onclick="runSrvice_Click"
    runat="server" Text="Execute"></asp:Button>
</p>
</form>
</body>
</html>
```

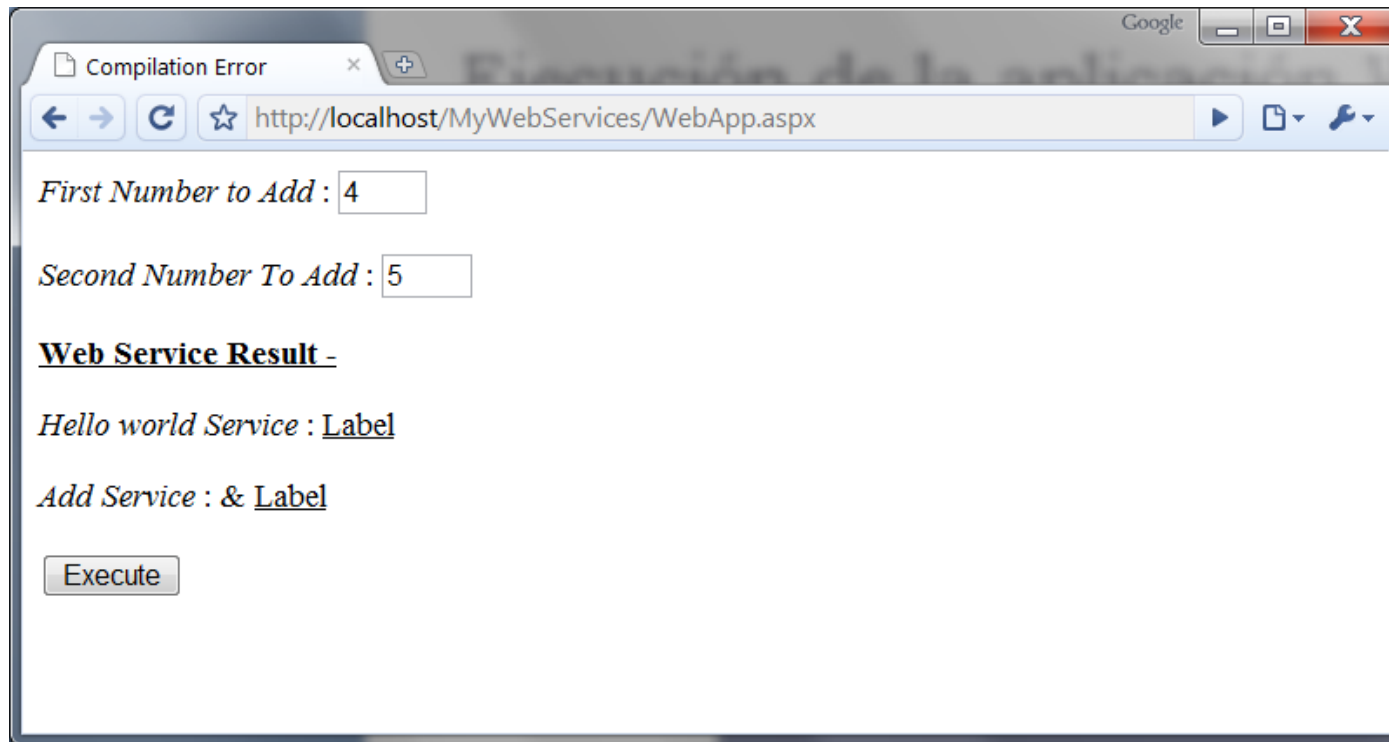
3.- Publicación del cliente (w7)

- ▶ Publicar el servicio:
 - ▶ Copiar a C:\inetpub\MywebServices el archivo c:\temp\WebApp.aspx
 - ▶ Confirmar como administrador



4.- Ejecución de la aplicación Web

- ▶ Abrir la página en navegador web:
 - ▶ <http://localhost/MyWebServices/WebApp.aspx>



Consumidor del servicio Web

Aplicación nativa

1. Creación del proxy
2. Creación del cliente nativo
3. Ejecución de la aplicación nativa

1.- Creación del proxy

▶ Preprocesado:

```
C:\temp> cmd.exe /V:ON /E:ON /K "C:\Program Files\Microsoft SDKs\ →  
→ windows\v6.1\bin\setenv.cmd" /Release  
C:\temp> WSDL http://localhost/MyWebServices/FirstService.asmx?WSDL
```

▶ Compilación:

```
C:\Temp\> csc /t:library FirstService.cs
```

2.- Creación del cliente nativo

▶ Código:

```
WinApp.cs
using System;
using System.IO;

namespace SvcConsumer {
    class SvcEater {
        public static void Main(String[] args) {
            FirstService mySvc = new FirstService();
            Console.WriteLine("Calling Hello World Service: " + mySvc.SayHello());
            Console.WriteLine("Calling Add(2, 3) Service: " + mySvc.Add(2, 3).ToString());
        }
    }
}
```

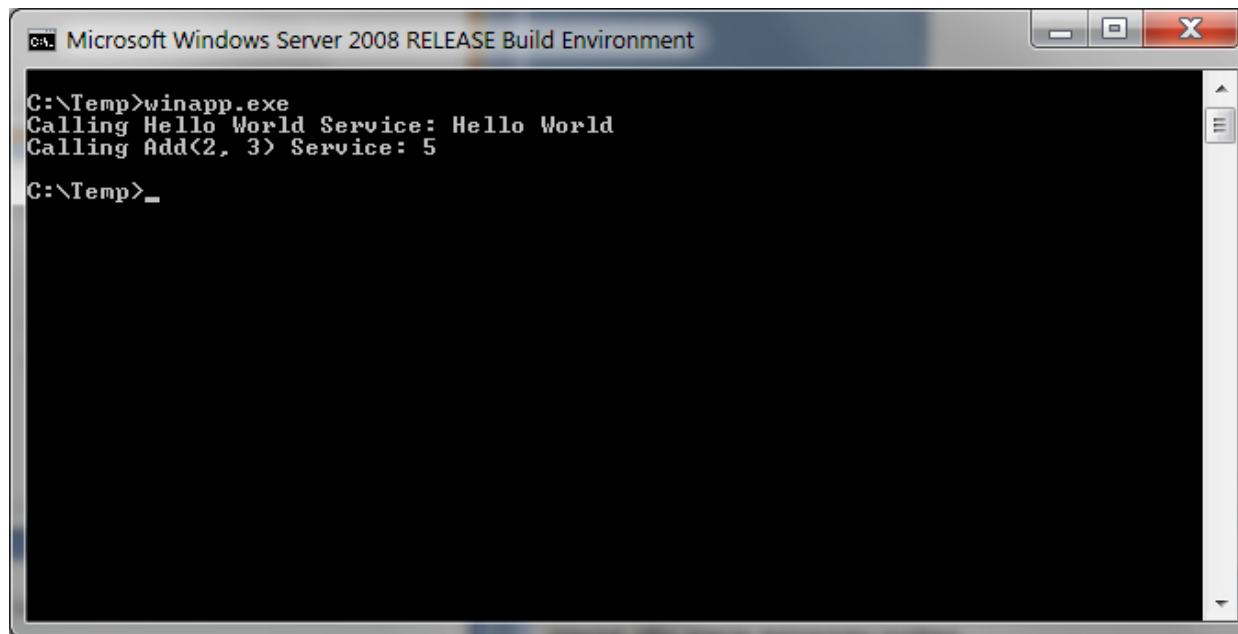
▶ Compilación:

```
C:\Temp> csc /r:FirstService.dll WinApp.cs
```


3.- Ejecución del cliente nativo

- ▶ Ejecutar como cualquier otra aplicación:

```
C:\Temp> WinApp.exe
```



```
Microsoft Windows Server 2008 RELEASE Build Environment
C:\Temp>winapp.exe
Calling Hello World Service: Hello World
Calling Add(2, 3) Service: 5
C:\Temp>_
```

Contenidos

1. Introducción Visual Studio .NET
2. Ejemplo con .NET SDK
3. Ejemplo con Visual Studio 2008

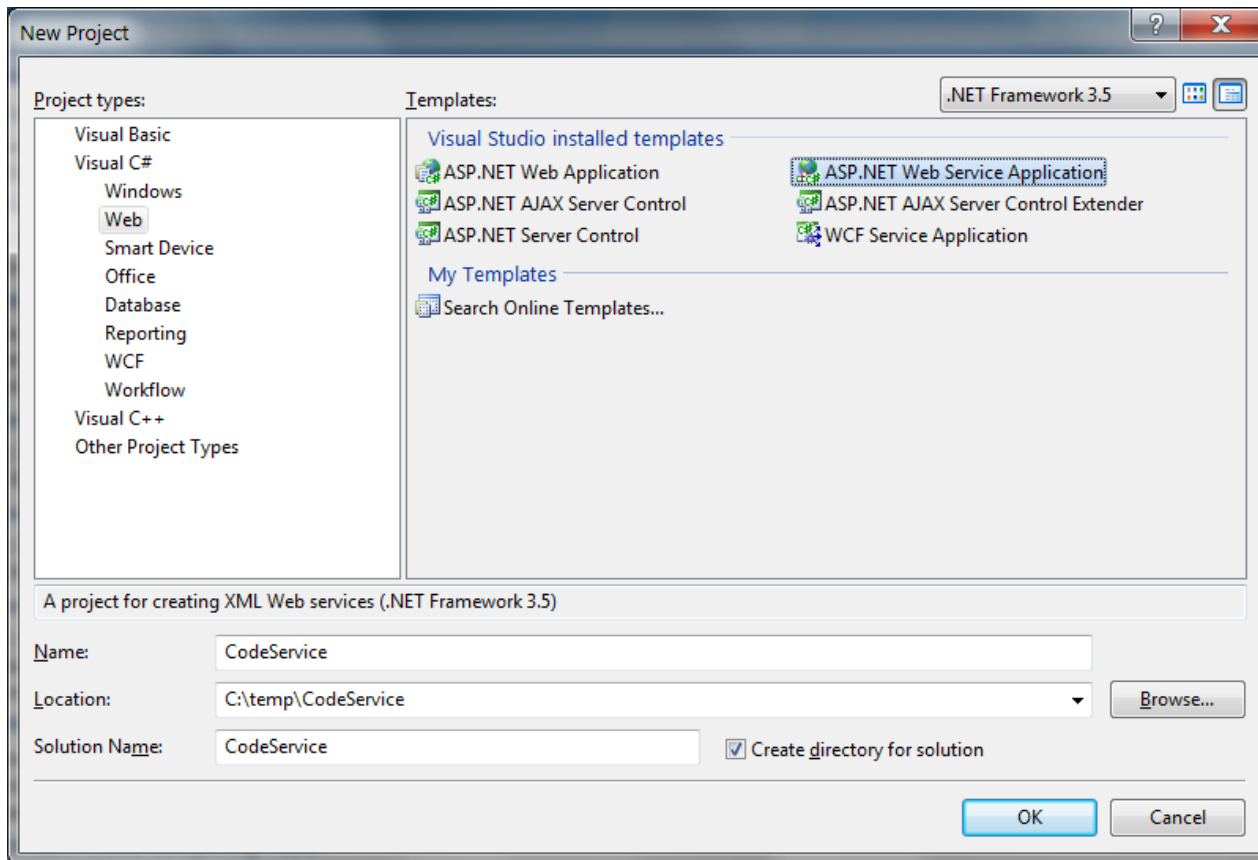


Servicio Web con Visual Studio

1. Creación del proyecto
2. Codificación del servicio Web
3. Comprobación de que está activo

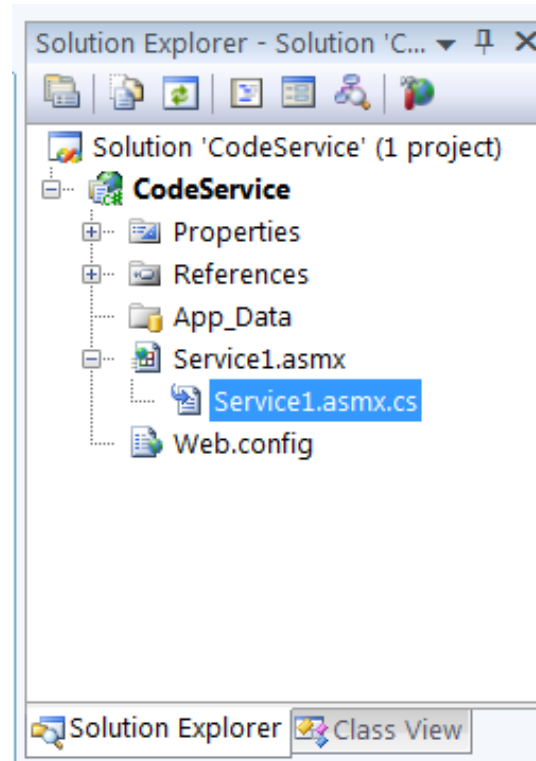
1.- Creación del proyecto

- ▶ Nombre del proyecto: **CodeService**



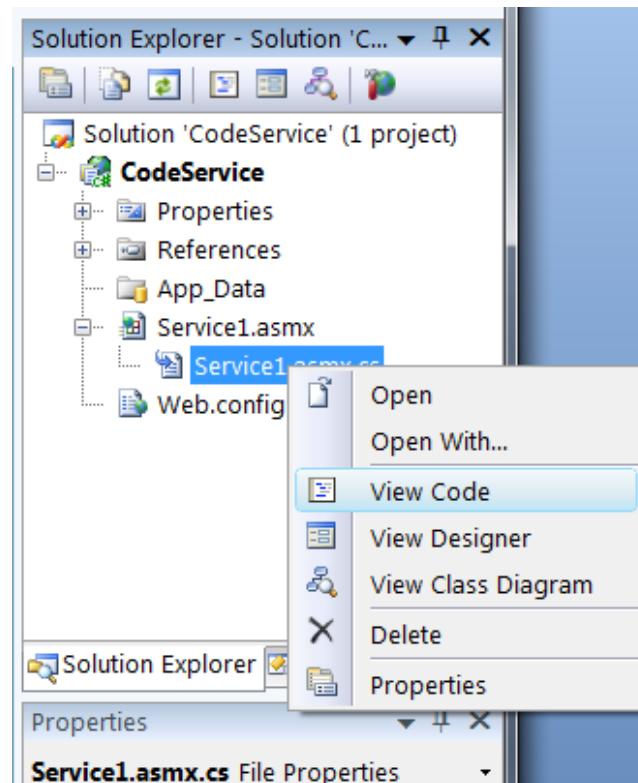
1.- Creación del proyecto

- ▶ Posible contenido inicial del proyecto:



1.- Creación del proyecto

- ▶ Pasar a *View Code*:



2.- Codificación del servicio Web

▶ Cambiar:

```
[WebMethod]
public string HelloWorld ()
{
    return "Hello World";
}
```

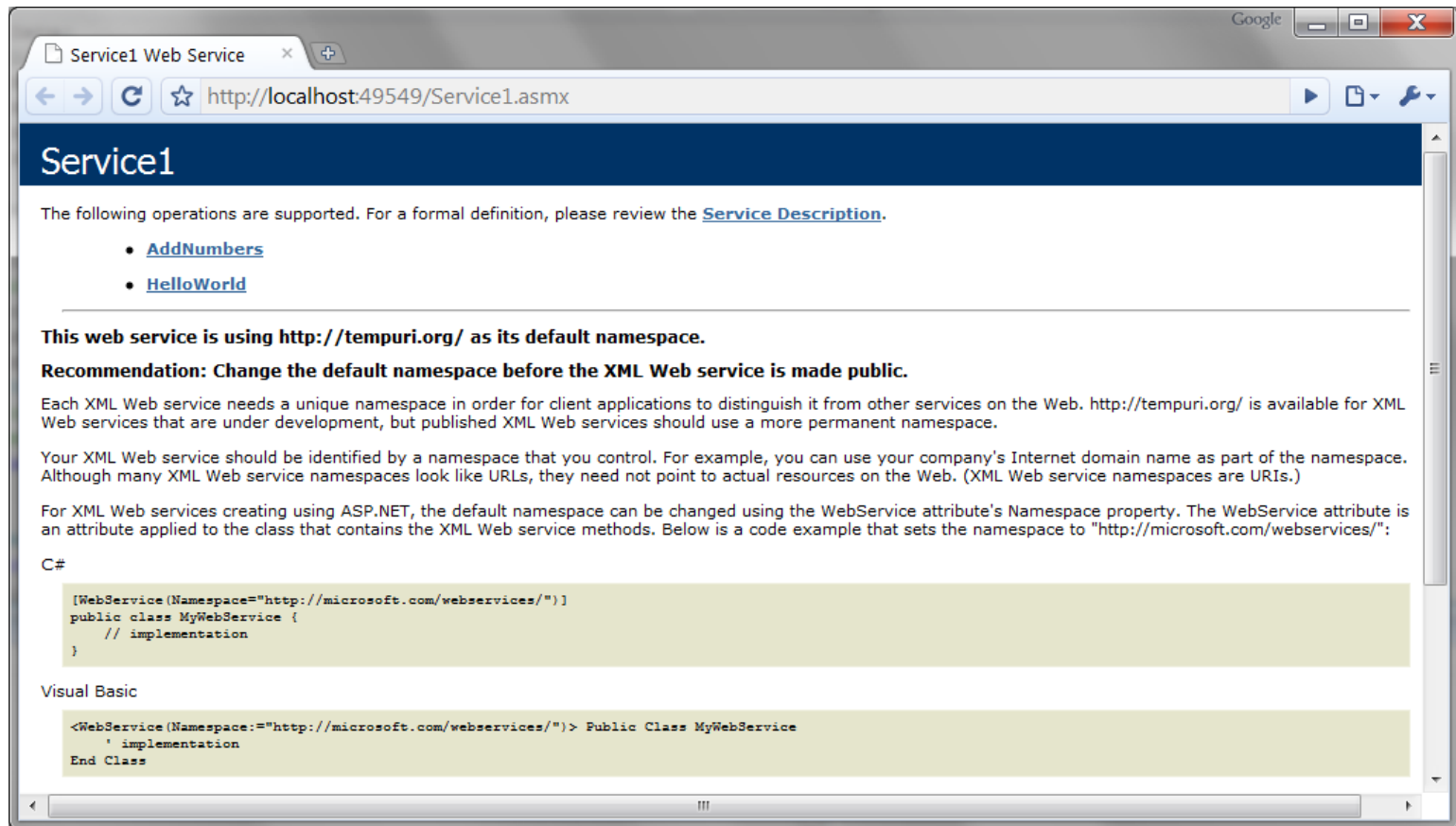
▶ Por:

```
[WebMethod]
public string HelloWorld(string lcName) {
    return "Hello World," + lcName;
}
```

```
[WebMethod]
public decimal AddNumbers(decimal lnNumber1, decimal lnNumber2) {
    return lnNumber1 + lnNumber2;
}
```

3.- Comprobación de que está activo

- ▶ Ejecutar el proyecto (tecla F5).



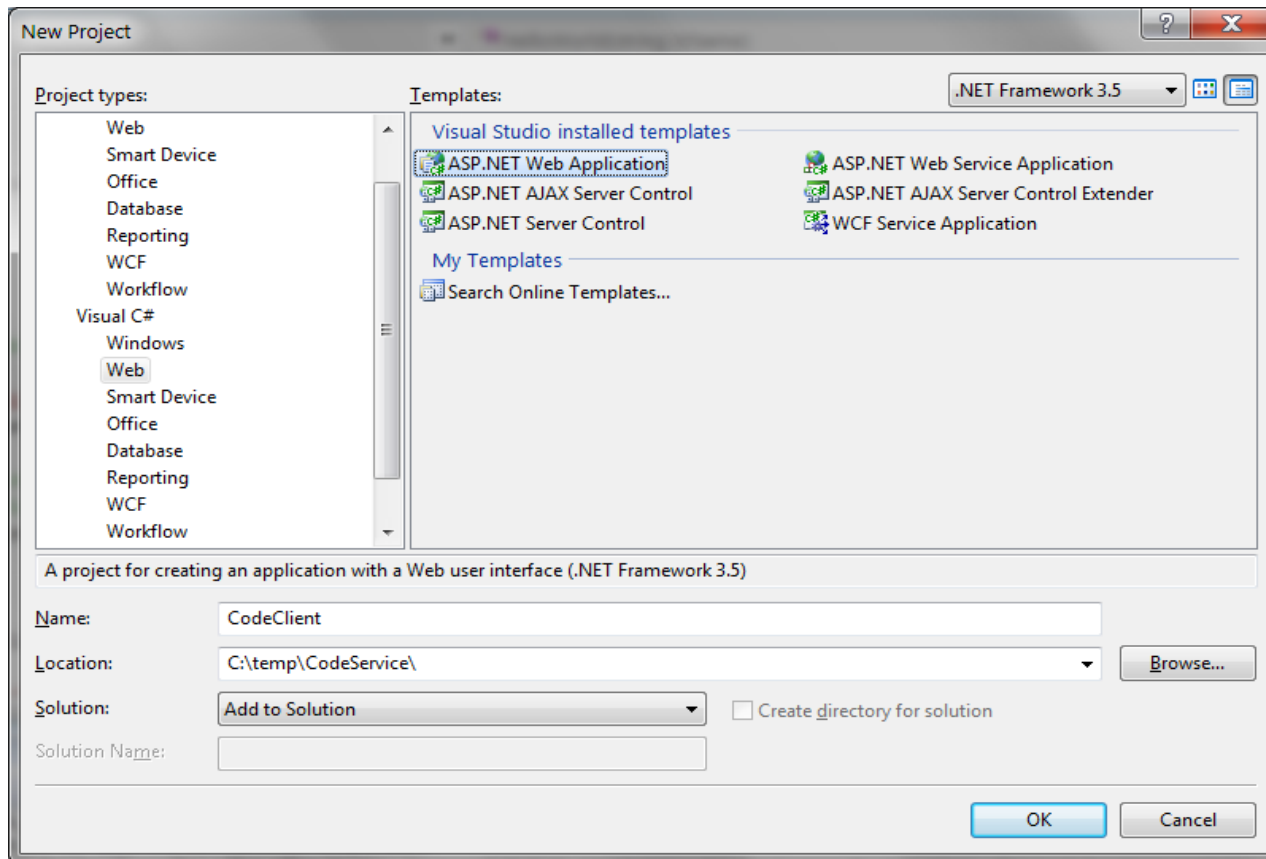
Cliente con Visual Studio

1. Creación del proyecto
2. Codificación del cliente
3. Ejecución del cliente



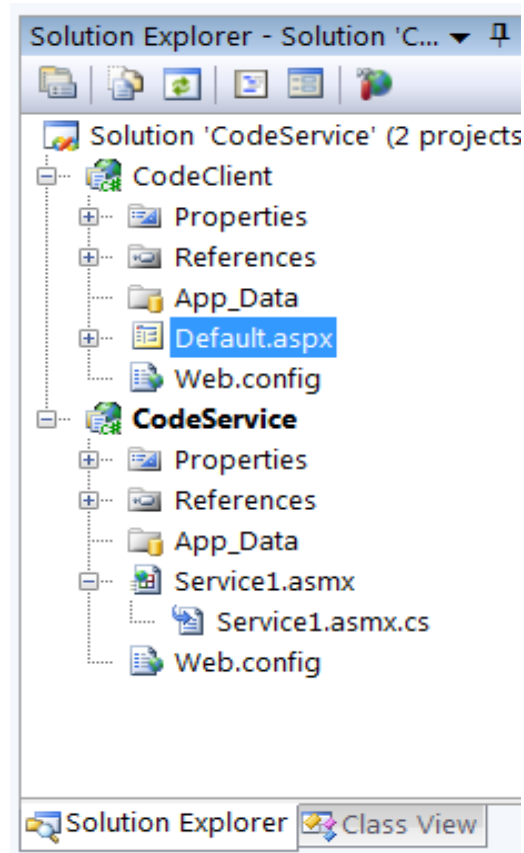
1.- Creación del proyecto

- ▶ Nombre del proyecto: **CodeClient**



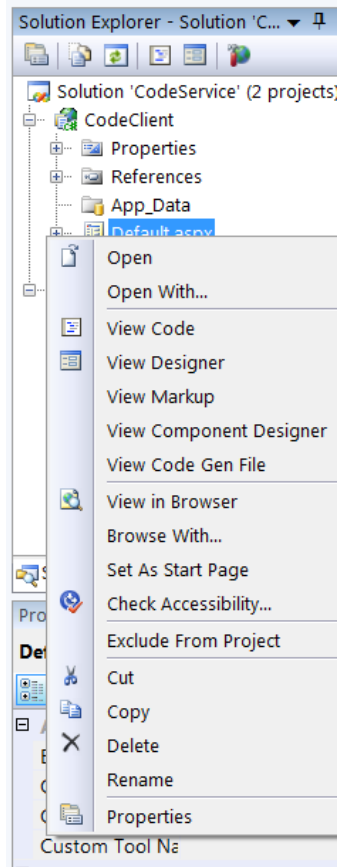
1.- Creación del proyecto

- ▶ Posible contenido inicial del proyecto:



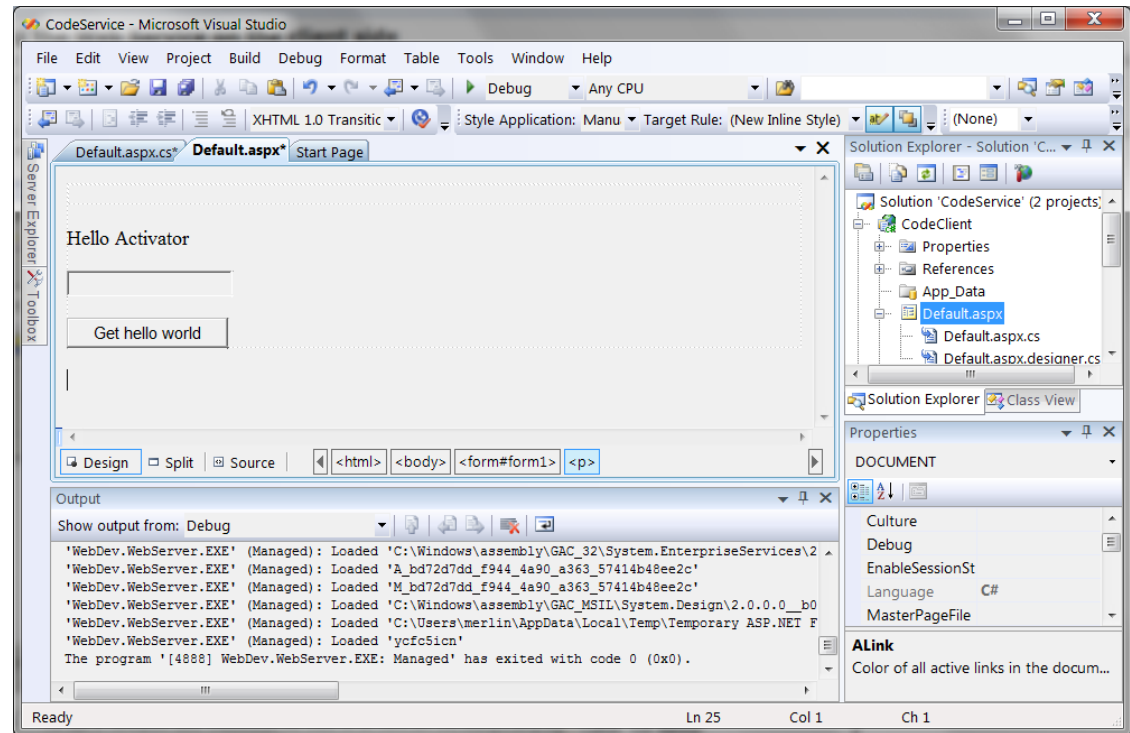
1.- Creación del proyecto

- ▶ Pasar a *View Designer*:



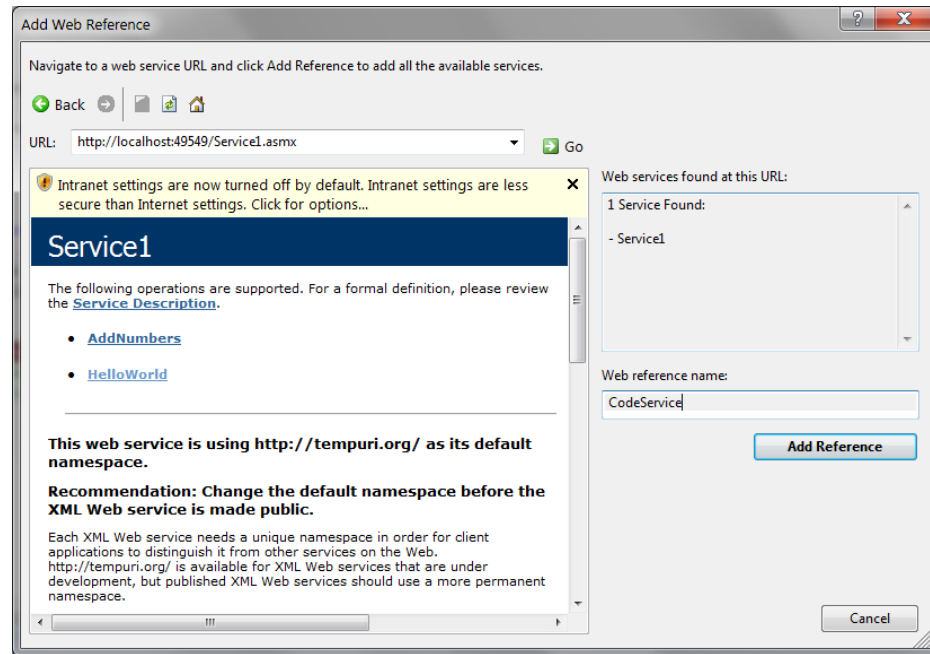
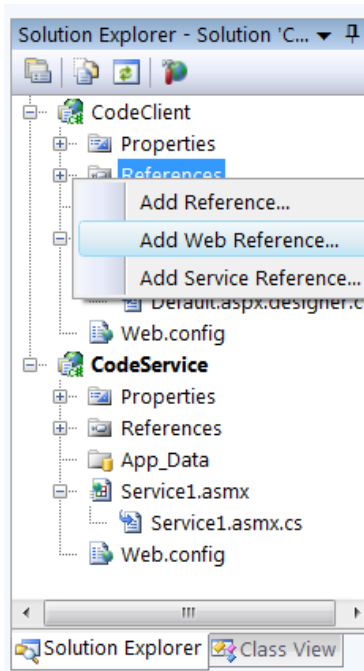
2.- Codificación del proyecto

- ▶ Diseñar la interfaz:
 - ▶ Añadir una caja de texto de nombre **TextBox1**
 - ▶ Añadir un botón de nombre **Button1**



2.- Codificación del proyecto

- ▶ Añadir referencia web:
 - ▶ Buscar el servicio en la solución



2.- Codificación del proyecto

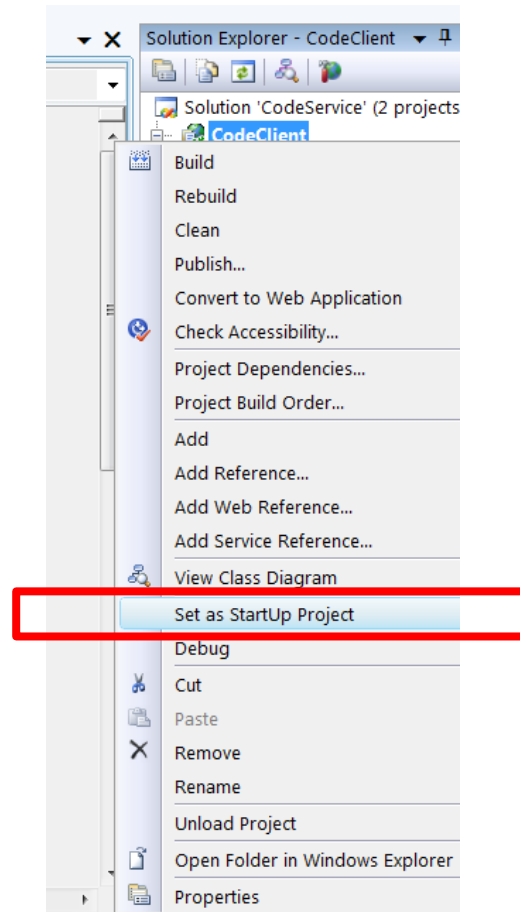
- ▶ Añadir la invocación al botón:

```
protected void Button1_Click(object sender, EventArgs e)
{
    CodeService.Service1 oService = new CodeService.Service1();

    TextBox1.Text = oService.HelloWorld("ARCOS.UC3M");
}
```

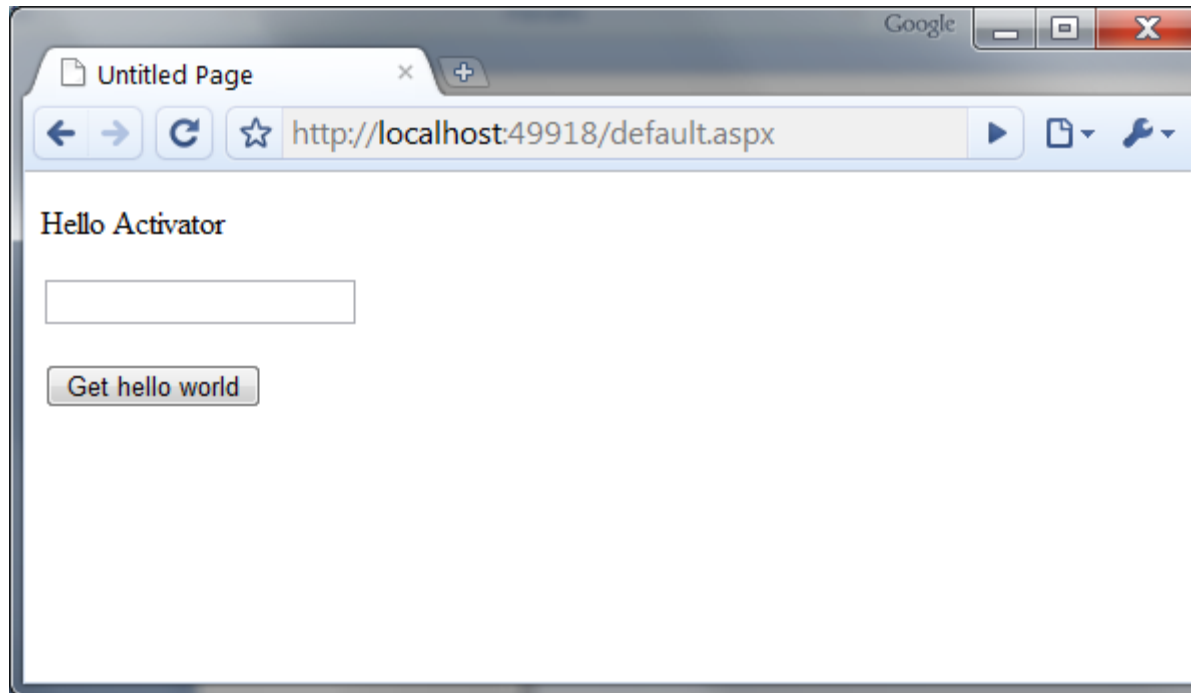
3.- Comprobación de que está activo

- ▶ Establecer **CodeClient** como proyecto de trabajo



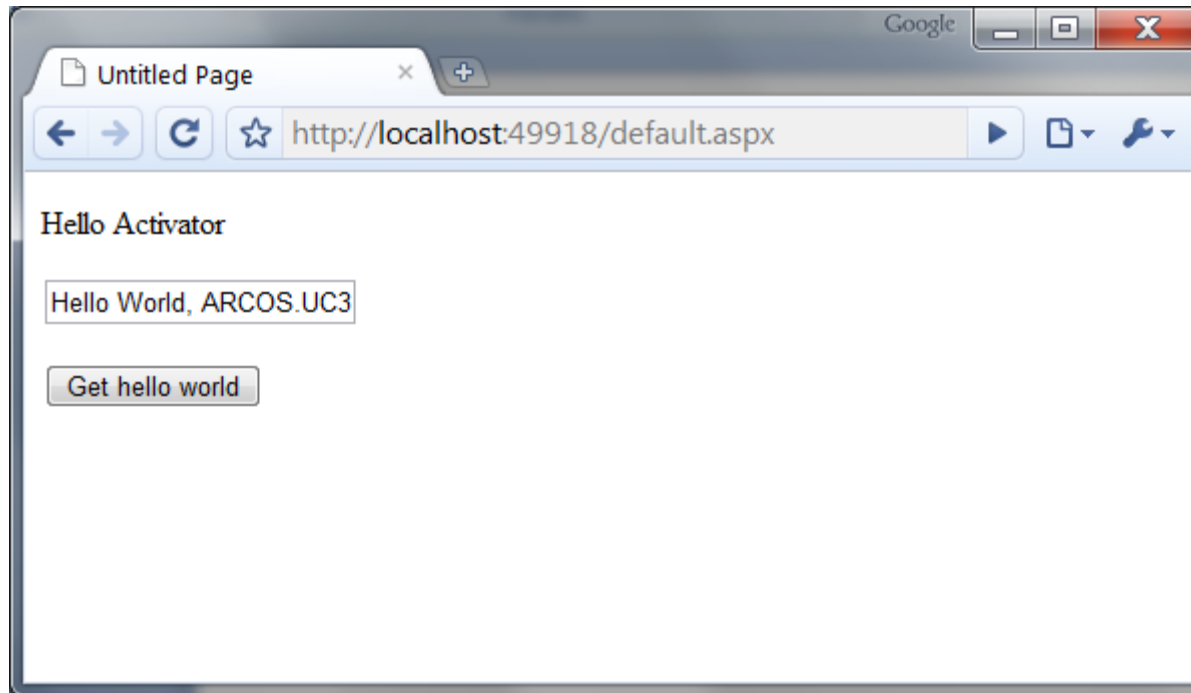
3.- Comprobación de que está activo

- ▶ Ejecutar el proyecto (tecla F5).



3.- Comprobación de que está activo

- ▶ Ejecutar el proyecto (tecla F5).





Desarrollo de aplicaciones distribuidas con .NET:
Servicios Web en .NET



Grupo ARCOS

Desarrollo de Aplicaciones Distribuidas

Ingeniería Informática

Universidad Carlos III de Madrid