

# El blog de Kañaz

Software, electrónica y IoT

HOME ACERCA DE

« Crear y consumir servicios web en Silverlight (Parte 1)

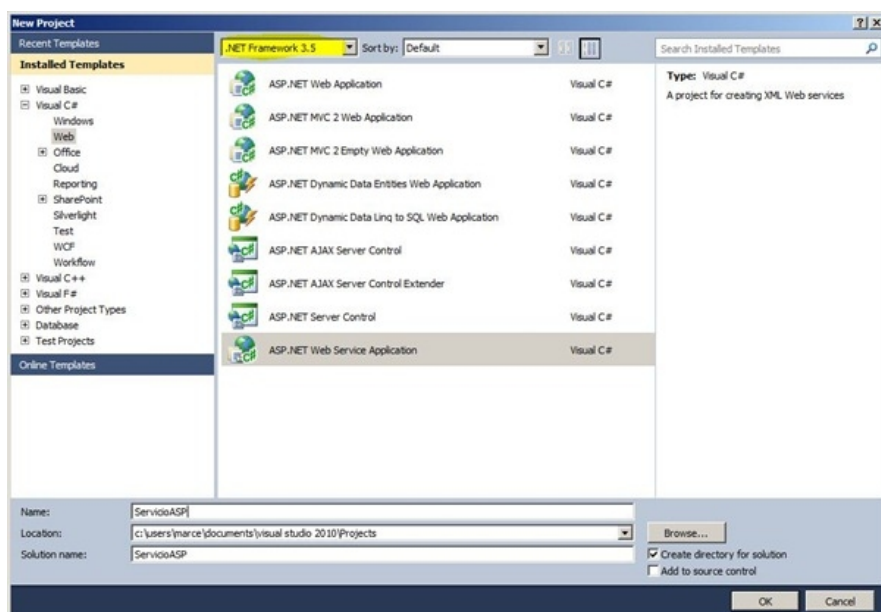
Crear y publicar una aplicación WCF (Parte 3) »

## Crear y publicar un servicio web ASP.NET (Parte 2)

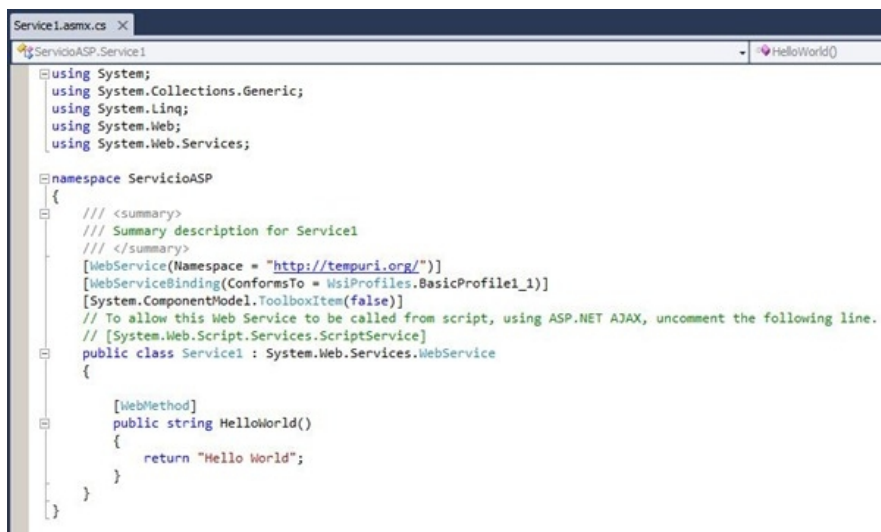
Twitter

En la primer entrada de esta mini serie de artículos relacionados vimos la introducción de la manera de como podemos vincular bases de datos con Silverlight, vamos a continuar con los servicios web ASP.NET.

Crea un nuevo proyecto en Visual Studio, de tipo Web Service Application ¡Aguas! Debes haber seleccionado el Framework 3.5.



Al aparecer el proyecto generado obtendrás una pantalla como la siguiente.



Busca en este sitio

Buscar ...

Meetup de Internet of things



Sígueme en Twitter

Seguir a @aminespinoza

Visita mis videos en Channel9



Channel 9

MI canal en Youtube

Escritorio luminoso RGB control...



Como puedes ver el método lleva antes el prefijo [WebMethod] y es público, debes considerar estos dos elementos para poder hacer tu método accesible desde la web, y es aquí en donde nos encargaremos de crear tres métodos, uno de consulta, uno de edición y uno de actualización (el de borrar quedará de tarea ¿Te late?).

**Antes de continuar, debo aclarar algo muy importante, seré un tanto injusto al crear estos servicios comparados con los WCF pero quiero hacerlo de la forma mas básica para que de paso puedas cubrir una gama mas amplia de tecnologías, pero para calmar mi conciencia debo decirte que puedes mezclar tecnologías sin problemas.**

Lo primero que insertaremos aquí será un método privado que nos permitirá el acceso a nuestra cadena de conexión.

```
private string cadenaConexion()
{
    string cadena = "Data Source=SERVIDORAMIN;Initial Catalog=Directorio;Integrated Security=True";
    return cadena;
}
```

## Consultar información

Este método nos dará toda la información de nuestros contactos.

```
[WebMethod]
public List<string> MostrarTodosLosContactos()
{
    List<string> listaClientes = new List<string>();
    SqlConnection conexion = new SqlConnection(cadenaConexion());
    SqlDataAdapter adaptador;
    string Consulta = "SELECT * FROM Contacto";
    DataTable datos = new DataTable();
    adaptador = new SqlDataAdapter(Consulta, conexion);
    adaptador.Fill(datos);

    foreach (DataRow registros in datos.Rows)
    {
        listaClientes.Add(registros["ID"].ToString() + "|" + registros["Nombre"].ToString() + "|" +
            registros["TELEFONO"].ToString() + "|" + registros["EMAIL"].ToString());
    }

    return listaClientes;
}
```

Creamos una lista de tipo string y por medio de los datos que obtenemos de la consulta vamos llenando la lista, así de simple.

## Insertar registros

Este método nos dará la posibilidad de insertar nuevos registros en nuestra tabla.

```
[WebMethod]
public int InsertarNuevaFecha(string Nombre, string Apellidos, string Telefono, string Email)
{
    string cadenaActualizar = "INSERT INTO Contacto (Nombre, APELLIDOS, TELEFONO, EMAIL) values (" +
        Nombre + ", " + Apellidos + ", " + Telefono + ", " + Email + ")";
    int t = 0;
    using (SqlConnection conexion = new SqlConnection(cadenaConexion()))
    {
        SqlCommand comando = new SqlCommand(cadenaActualizar, conexion);
        conexion.Open();
        comando.Parameters.AddWithValue("@nombre", Nombre);
        comando.Parameters.AddWithValue("@apellidos", Apellidos);
        comando.Parameters.AddWithValue("@telefono", Telefono);
        comando.Parameters.AddWithValue("@email", Email);
        t = comando.ExecuteNonQuery();
        conexion.Close();
    }
    return t;
}
```

Como puedes ver, lo que hacemos es simplemente recibir los parámetros y en base a ellos insertar un nuevo elemento en nuestra tabla, como es obligatorio que TODOS los métodos accesibles desde web regresen un valor, en este caso regresamos el valor de las operaciones realizadas.

## Actualizar registros

Este método nos dará la posibilidad de modificar a los registros que ya tenemos, ya sea en uno solo o en todos sus atributos.

```
[WebMethod]
public int ActualizarContacto(string ID, string Nombre, string Apellidos, string Telefono, s
{
    string cadenaActualizar = "UPDATE Contacto SET Nombre=@nombre, APELLIDOS=@apellidos, TELEFON
    int t = 0;
    using (SqlConnection conexion = new SqlConnection(cadenaConexion()))
    {
        SqlCommand comando = new SqlCommand(cadenaActualizar, conexion);
        conexion.Open();
        comando.Parameters.AddWithValue("@nombre", Nombre);
        comando.Parameters.AddWithValue("@apellidos", Apellidos);
        comando.Parameters.AddWithValue("@telefono", Telefono);
        comando.Parameters.AddWithValue("@email", Email);
        t = comando.ExecuteNonQuery();
        conexion.Close();
    }
    return t;}
```

Aquí de una forma muy parecida al método anterior recibimos todos los argumentos y con la sentencia SQL actualizamos los datos en función de su campo ID.

Bien, si ya tienes los métodos completados y ejecutas tu servicio web, el resultado será muy similar a esto.

Service1

Las operaciones siguientes son compatibles. Para una definición formal, revise la [descripción de servicios](#).

- ActualizarContacto
- InsertarNuevaFecha
- MostrarTodosLosContactos

---

Este servicio Web utiliza <http://tempuri.org/> como espacio de nombres predeterminado.

**Recomendación:** cambiar el espacio de nombres predeterminado antes de hacer público el servicio Web XML.

Cada servicio Web XML necesita un espacio de nombres único para que las aplicaciones de cliente puedan distinguir este servicio de otros servicios del Web permanente.

Debe identificar su servicio Web XML con un espacio de nombres que controle. Por ejemplo, puede utilizar el nombre de dominio de Internet de su compañía Web. (Los espacios de nombres de los servicios Web XML son los URI.)

En los servicios Web XML que se crean con ASP.NET, se puede cambiar el espacio de nombres predeterminado utilizando la propiedad Namespace del atributo establece el espacio de nombres en "http://microsoft.com/webservices/":

```
C#
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementación
}
```

Vamos a hacer una pequeña prueba y seleccionar el método MostrarTodosLosContactos, veras una pantalla así.

Service1

Haga clic [aquí](#) para obtener una lista completa de operaciones.

---

MostrarTodosLosContactos

**Prueba**

Haga clic en el botón "Invocar", para probar la operación utilizando el protocolo HTTP POST.

Invocar

---

**SOAP 1.1**

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.1. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/MostrarTodosLosContactos"
```

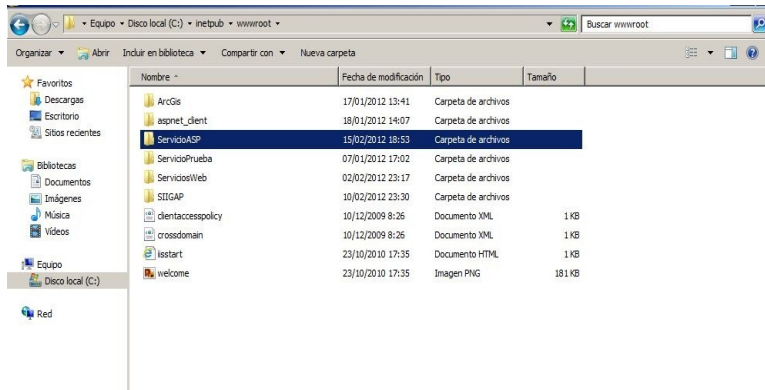
Si pruebas en los otros dos métodos notarás que te piden que llenes información, esto es debido a los parámetros que los métodos necesitan para poder operar de manera adecuada. En el caso del método que usamos, presiona el botón Invocar y tu resultado debe ser este.

```
<?xml version="1.0" encoding="UTF-8"?>
- <ArrayOfString xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <string>2|Carlos Amín|Espinoza de los Monteros|5550468524|amin.espinosa@gmail.com</string>
  <string>3|Marcela|Martínez Quiroga|5548505600|marce@hotmail.com</string>
  <string>4|Geovanni|Sanchez Rodriguez|7757845124|geovanni@hotmail.com</string>
  <string>5|Felipe|Bello Jardines|7717941008|felipe@hotmail.com</string>
</ArrayOfString>
```

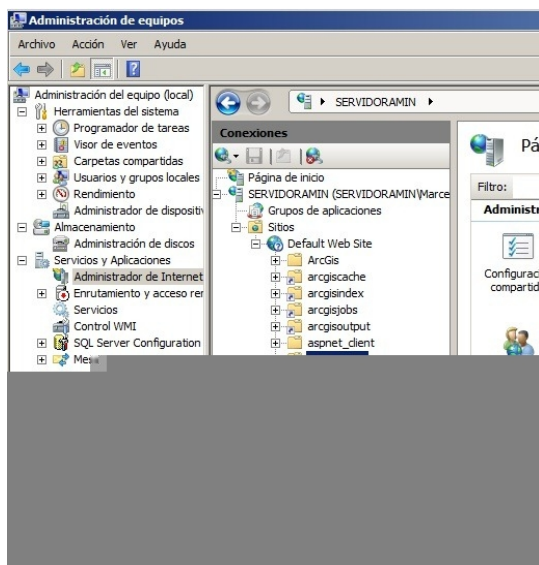
Así es, nada mas que un XML que nos estará regresando la información que le solicitamos. Hasta aquí ya tenemos nuestro servicio web realizado, pero para que Silverlight pueda consumirlo, deberemos publicarlo en nuestro IIS, así que vamos para eso.

## Publicar mi servicio en IIS

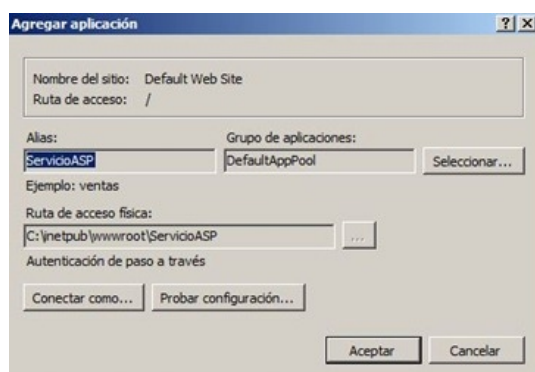
En la ruta C:\inetpub\wwwroot (debes tener instalado IIS para esto), crea una nueva carpeta.



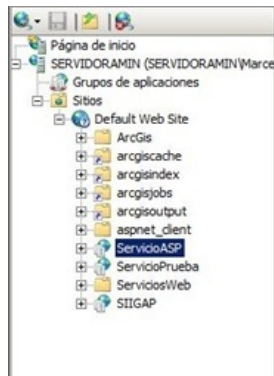
Ya con la carpeta creada, vete al administrador de tu equipo. (Presiona la tecla de Windows + R y escribe el comando `compmgmt.msc`), selecciona al administrador de IIS, expande los nodos de la derecha hasta llegar a la posición de la siguiente imagen.



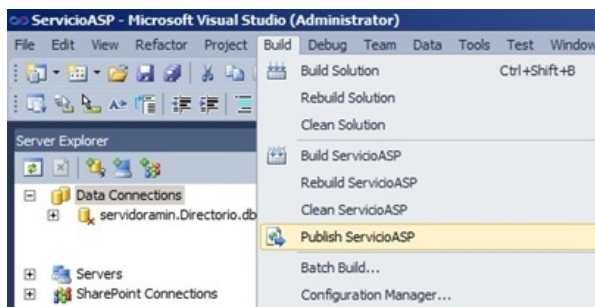
Al seleccionar la opción marcada te aparecerá una ventana como la de abajo, solo da clic en aceptar.



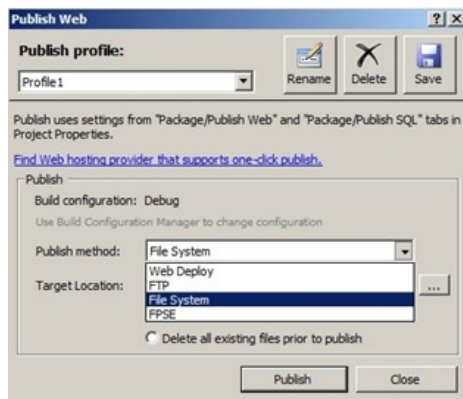
Y ahora verás que tu antigua carpeta se verá de la siguiente forma.



Ahora podremos regresar a Visual Studio y seleccionar en el menú de "Build" la opción de "Publish Service".



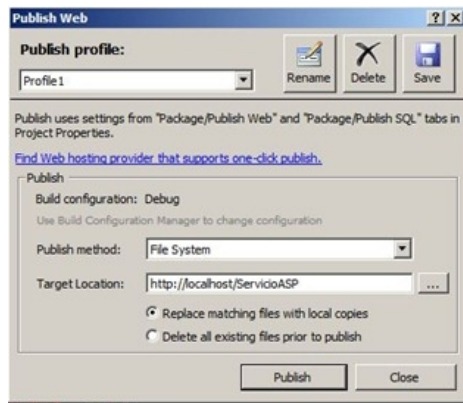
En el cuadro de diálogo escoge la opción de File System en la opción de Publish method y en la de Target Location presiona el botón de "Examinar" (el de los tres puntitos pues).



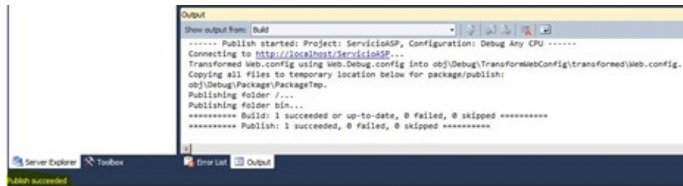
Al presionar el botón de examinar, tendrás en pantalla ahora la siguiente ventana (si no te apareció es porque no estás corriendo tu Visual Studio en modo administrador), ahí simplemente selecciona la carpeta que acabas de crear.



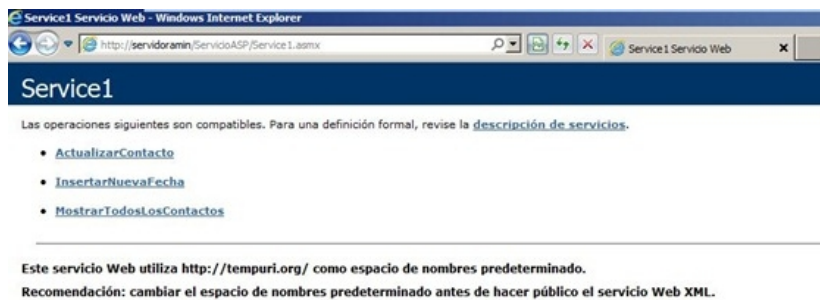
Tu cuadro de diálogo inicial quedará de la siguiente forma.



Solo basta que presiones "Publish" y veas en la esquina inferior izquierda y en tu ventana de Output los siguientes mensajes.



Ya que tienes ese letrero, puedes escribir la url de tu sitio con la siguiente secuencia `http://NombreDeTuServidor/NombreDeTuCarpeta/NombreDeTuServicio`, en mi caso quedó así "`http://localhost/ServiceASP/Service1.aspx`".



¡Listo! Ya tienes tu servicio web publicado. En el artículo cuatro de esta serie te mostraré como consumir este servicio web en tu aplicación, por ahora solo prueba los métodos, estos deben funcionar exactamente igual que en tu aplicación en modo de desarrollo ahora que ya esta en modo productivo.

Descarga aquí el código de ejemplo.

Posted on 16 febrero, 2012 by Amin Espinoza. This entry was posted in Servicios Web/Datos, Silverlight, Tutoriales, WCF. Bookmark the permalink.

« Crear y consumir servicios web en Silverlight (Parte 1)

Crear y publicar una aplicación WCF (Parte 3) »

## 15 Comments



hyl

29 mayo, 2012 - 3:00 am | Permalink

ya tengo mi sitio en el servidor local iis , corre perfecto el localhost, pero e tratado de ponerlo en un servidor externo y cuando accedo a el me aparece solo codigo que estoy asiendo mal ,

Responder



Facundo

27 septiembre, 2012 - 6:46 am | Permalink

Hola, veo que al principio creas el método "MostrarTodosLosContactos()" que devuelve una lista de tipo "List", pero la clase "List" no esta creada. Como lo soluciono?

Responder

Amin Espinoza





27 septiembre, 2012 - 2:41 pm | Permalink

Estás completamente cierto Facundo, la lista debe ser tipo "string", en un minuto lo corregiré en el ejemplo, gracias por la observación

Responder

**Facundo**

29 septiembre, 2012 - 1:25 am | Permalink

Muchas gracias, ahora puedo seguir. Muy bueno el tutorial, por cierto.

Responder

**Pintor**

27 abril, 2013 - 7:36 pm | Permalink

Hola que tal hice lo que dicen aqui en la informacion pero me tira el siguiente error.  
Error HTTP 500.19 – Internal Server Error  
No se puede obtener acceso a la página solicitada porque los datos de configuración relacionados de la página no son válidos.

Error de configuración Hay definida una sección 'system.web.extensions/scripting/scriptResourceHandler' duplicada

Responder

**Liz**

29 abril, 2013 - 2:33 pm | Permalink

Interesante tu tutorial, pero me gustaria saber como hacer para publicar el servicio web desde código, lo que pasa es que estoy haciendo un instalador que copia la carpeta en el inetpub wwwroot, pero ahora necesito que se publique, he buscado, pero hasta ahora no he encontrado como automatizar este proceso para que el usuario "no haga nada". Si me pudieras guiar, te agradecería mucho.

Responder

**Mariano**

2 julio, 2013 - 12:57 am | Permalink

Hola, hice el servicio, me funciona perfecto cuando lo ejecuto desde el VS, pero cuando lo publiqué, no me apareció la página y solo me aparece:

Tenés idea de por qué puede ser?  
Lo mismo me pasa con una página que tengo en el IIS.  
Si lo abro desde el IE, lo abre y me muestra eso, si lo abro desde chrome, me descarga la página...  
Toda información será bienvenida.  
Saludos.

Responder

**José Durazo**

24 abril, 2014 - 9:33 pm | Permalink

GRACIAS, una explicación simple que no se encuentra facilmente

Responder

**AnzOne**

14 mayo, 2014 - 4:37 pm | Permalink

Muchas gracias, buena la explicación con los pantallazos.

Responder

**Paty**

12 marzo, 2015 - 9:28 pm | Permalink

Buen día, estuve revisando tu tutorial, quería ver si alguien me puede apoyar con este problema.

Tengo un web service que corre perfectamente el local, pero al publicarlo en el servidor me manda el siguiente error:

This page contains the following errors:

error on line 1 at column 2: StartTag: invalid element name  
Below is a rendering of the page up to the first error.

Alguien me puede decir como solucionar el problema, leí algunos blocqs sobre este error pero no he podido solucionar el problema.

Gracias

Responder



**Hansel Tribeños Ibarra**

2 julio, 2015 - 4:58 pm | [Permalink](#)

Una consulta a mi no me aparece el campo aspnet\_client en la lista de default web site

Responder



**Jose Aaron**

20 agosto, 2015 - 10:49 pm | [Permalink](#)

Hola!, ¿Que tal?

Tengo una duda...

Realice un servicio web y lo intento consumir en android, mi servicio web contiene un método para insertar 2 números en una base de datos en SQL Server , cuando ejecuto mi servicio web realiza los registros, pero cuando intento publicar mi servicio web y accedo a el me manda un error de acceso al servidor SQL, anteriormente ya había consumido un servicio web en android pero en MySql y no se cual podría ser el error en esta ocasión, podrías ayudarme por favor?

Responder



**Amin Espinoza**

30 noviembre, 2015 - 10:59 pm | [Permalink](#)

Este tipo de servicios son SOAP y difícilmente los podrás consumir en Android, lo que debes hacer es crear un servicio tipo REST para poder moverte sin problemas y no se te dificulte la conexión.

Responder



**Maikel Cutiño Fernandez**

11 enero, 2017 - 3:43 am | [Permalink](#)

Hola quisiera saber como consumir este servicio web

Responder



**Amin Espinoza**

12 enero, 2017 - 2:03 am | [Permalink](#)

Lo haces por medio de SOAP

Responder

## Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con \*

### Comentario

Nombre \*

Correo electrónico \*



Web

**Publicar comentario**

This blog is kept spam free by WP-SpamFree.

A nice revamping of Sandbox theme for WordPress