

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

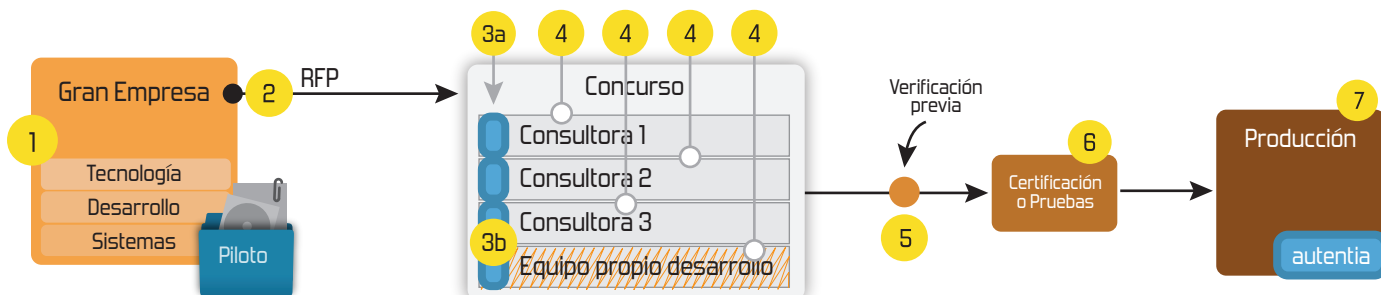
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



-¿Crees que la informática es sólo programar?
-¿Sabrías organizar eficientemente un equipo?
-¿Quién te resuelve las dudas sobre esta profesión?..... o te siembra más?

Hosting patrocinado por **enREDados**



Inicio **Quienes somos** **Tutoriales** **Formación** **Comparador de salarios** **Comentar libro** **Charlas** **Más**

Estas en: **Inicio** **Tutoriales** Contract-First web services con Visual Studio 2008

Últimas Noticias

- » Formación por calendario
- » El próximo día 22 a las 19:30 vamos a celebrar el primer Coding Dojo organizado por agilismo.es
- » Comentando el libro: Eneagrama y éxito personal de Ginger Lapid-Bogda
- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Disponible la primera versión de los plugins para integrar Maven y Bugzilla.
- » Lanzamiento del nuevo Web de Autentia

+Noticias Destacadas

- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Autentia patrocina el primer Agile Open Spain
- » Lanzamiento del nuevo Web de Autentia
- » Autentia cumple 6 años

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Tutorial desarrollado por



Iván García Puebla

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/JEE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [webservice-contract-first-vs2008.pdf](#)

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en [AdictosAlTrabajo...](#)

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)

Fecha de creación del tutorial: 2009-12-11

Contract-First web services con Visual Studio 2008

Índice

1. Introducción
2. Descriptor del contrato
 1. Definición de los tipos del documento con XML Schema
 2. Definición del contrato con WSDL
3. Implementación con Visual Studio 2008. WSCF.blue
 1. Instalación de WSCF.blue
 2. Crear el servicio e importar el contrato
 3. Implementar el web service
 4. Publicar y acceder al servicio
4. Probar el web service con soapUI
5. Conclusión

Introducción

En el diseño y desarrollo de servicios web podemos utilizar dos aproximaciones:

- Dirigido por contrato, o **Contract-First**. Primero se describen las funcionalidades del servicio web en cuanto a su interfaz, semántica y aspectos administrativos de la invocación, siendo la **especificación** WSDL el estándar. El WSDL, como documento XML, puede ser parseado por una herramienta para facilitar la implementación del web service proveedor o consumidor.
- **Contract-Last**, en el que primero se implementa el servicio web en un determinado lenguaje y posteriormente se genera un WSDL que lo describa.

Cada cual tiene sus [ventajas e inconvenientes](#) [1] [2]. En este tutorial veremos la primera de las dos opciones, implementando en C# un servicio web proveedor. Utilizaremos las siguientes herramientas:

- Windows Vista Home Edition SP2, C2D@1.5, 3GB Ram
- Microsoft Visual Studio 2008 Professional SP1
- Microsoft .NET Framework 3.5 SP1
- thinktercure WSCF.blue V1 Final (1.0.5) (*)
- soapUI 3.0.1

(*) En el momento de escribir este tutorial no existe una versión para [Visual Studio 2008 Express Edition](#)

El código fuente de este tutorial puede descargarse aquí: [wsEncriptacion_adictosaltrabajo.zip](#)

Descriptor del contrato

Vamos a trabajar con un web service de encriptación mediante [Triple-DES](#) y [MD5](#) que ofrece dos operaciones :

- **encriptar**. Recibe tres tipos simples: la cadena de texto en claro, la clave de encriptación y un flag para el uso o no de hashing. Devuelve la cadena de texto encriptada.
- **desencriptar**. Recibe un tipo complejo compuesto por tres tipos simples: la cadena de texto encriptada, la clave utilizada en la encriptación y el flag para indicar el uso de hashing. Devuelve la cadena de texto desencriptada.

El uso de tipos simples o complejos es una cuestión de diseño, y en nuestro caso lo hacemos así para ampliar la cobertura de capacidades de WSCF.blue.

Definición de los tipos del documento con XML Schema



☐ Web

☒ [www.adictosaltrabajo.com](#)

Últimos tutoriales

2009-12-11
[Contract-First web services con Visual Studio 2008](#)

2009-12-09
[Integrando Sonar con Hudson](#)

2009-12-09
[Apache + Tomcat: Balanceo de carga y alta disponibilidad](#)

2009-12-08
[MySQL: Replicación de bases de datos en MySQL](#)

2009-12-07
[Analizando la calidad del código Java con Sonar](#)

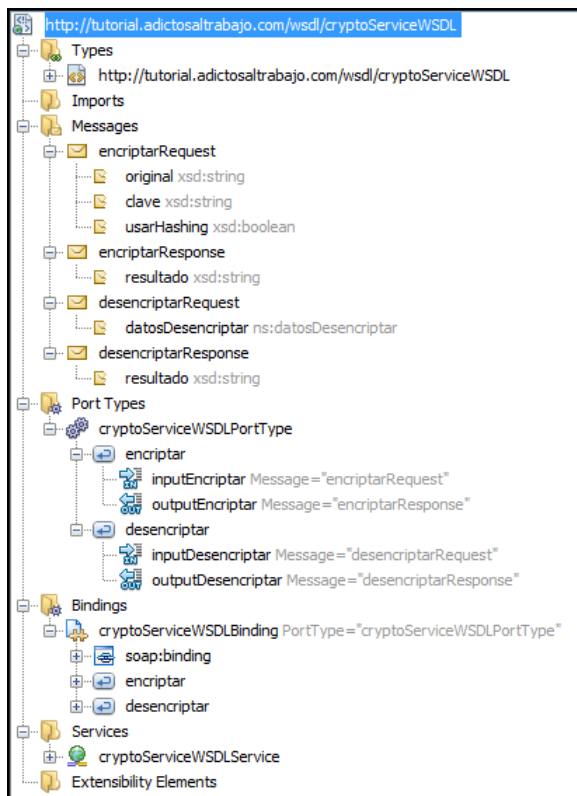
2009-12-03
[Instalar OpenESB 2.1 e Introducción](#)

La definición del tipo de datos complejo lo implementamos mediante un XML Schema. Se muestra a continuación, y puede descargarse aquí: [cryptoSchema.xsd](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  cryptoSchema.xsd
  Tutorial: Contract-First web services con Visual Studio 2008
  www.adictosaltrabajo.com - Ivan Garcia Puebla
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://tutorial.adictosaltrabajo.com/schema/cryptoSchema"
  xmlns:tns="http://tutorial.adictosaltrabajo.com/schema/cryptoSchema"
  elementFormDefault="qualified">
  <xsd:complexType name="datosDesencriptar">
    <xsd:sequence>
      <xsd:element name="encriptado" type="xsd:string"/></xsd:element>
      <xsd:element name="clave" type="xsd:string"/></xsd:element>
      <xsd:element name="usarHashing" type="xsd:boolean"/></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Definición del contrato con WSDL

Implementamos el contrato del web service en un descriptor WSDL concreto. Utilizaremos el Schema anterior, SOAP sobre HTTP, y el resto de detalles se muestran a continuación:



descriptor cryptoService.wsdl

El fuente del fichero puede descargarse aquí: [cryptoService.wsdl](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  cryptoServiceWSDL.wsdl
  Tutorial: Contract-First web services con Visual Studio 2008
  www.adictosaltrabajo.com - Ivan Garcia Puebla
-->
<definitions name="cryptoServiceWSDL"
  targetNamespace="http://tutorial.adictosaltrabajo.com/wsdl/cryptoServiceWSDL"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://tutorial.adictosaltrabajo.com/wsdl/cryptoServiceWSDL"
  xmlns:ns="http://tutorial.adictosaltrabajo.com/schema/cryptoSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types>
    <xsd:schema targetNamespace="http://tutorial.adictosaltrabajo.com/wsdl/cryptoServiceWSDL">
      <xsd:import namespace="http://tutorial.adictosaltrabajo.com/schema/cryptoSchema"
        schemaLocation="cryptoSchema.xsd"/>
    </xsd:schema>
  </types>
  <message name="encriptarRequest">
    <part name="original" type="xsd:string"/>
    <part name="clave" type="xsd:string"/>
    <part name="usarHashing" type="xsd:boolean"/>
  </message>
```

2009-11-25
[Tutorial de Google Forms](#)

2009-11-23
[Alfresco - Modificando y eliminando contenido desde nuestras aplicaciones Java](#)

2009-11-23
[Alfresco - Añadiendo contenido desde nuestras aplicaciones Java](#)

2009-11-23
[Haciendo funcionar Google Chromium Operating System sobre Windows XP](#)

2009-11-23
[Redimensionar el tamaño de una partición de VirtualBox](#)

2009-11-20
[El Arte de las Presentaciones. Siguiendo la Senda Zen.](#)

2009-11-18
[Tutorial basico de google wave bots](#)

2009-11-13
[Introducción a Escritorios Animado \(Winamp y MilkWave \)](#)

2009-11-12
[Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.](#)

2009-11-12
[Tutorial básico sobre Google Application Engine](#)

2009-11-11
[Instalación de Glassfish 2.1](#)

2009-11-26
[Rational Software Architect y Rational RequisitePro](#)

2009-10-27
[Cómo conseguir que Subversion avise a Hudson para lanzar una build](#)

2009-10-26
[Cómo instalar Hudson en Apache Tomcat](#)

2009-10-26
[Estilos externos en iReport](#)

2009-10-14
[Echo en MySQL](#)

2009-09-28
[iReport : Solución al problema de los parámetros dinámicos.](#)

2009-09-27
[Mavenizar Liferay SDK](#)

2009-09-27
[Crear un plugin con Liferay SDK](#)

```

<message name="encryptarResponse">
  <part name="resultado" type="xsd:string"/>
</message>
<message name="desencryptarRequest">
  <part name="datosDesencryptar" type="ns:datosDesencryptar"/>
</message>
<message name="desencryptarResponse">
  <part name="resultado" type="xsd:string"/>
</message>
<portType name="cryptoServiceWSDLPortType">
  <operation name="encryptar">
    <input name="inputEncryptar" message="tns:encryptarRequest"/>
    <output name="outputEncryptar" message="tns:encryptarResponse"/>
  </operation>
  <operation name="desencryptar">
    <input name="inputDesencryptar" message="tns:desencryptarRequest"/>
    <output name="outputDesencryptar" message="tns:desencryptarResponse"/>
  </operation>
</portType>
<binding name="cryptoServiceWSDLBinding" type="tns:cryptoServiceWSDLPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="encryptar">
    <soap:operation/>
    <input name="inputEncryptar">
      <soap:body use="literal"
        namespace="http://tutorial.adictosaltrabajo.com/wsd/cryptoServiceWSDL"/>
    </input>
    <output name="outputEncryptar">
      <soap:body use="literal"
        namespace="http://tutorial.adictosaltrabajo.com/wsd/cryptoServiceWSDL"/>
    </output>
  </operation>
  <operation name="desencryptar">
    <soap:operation/>
    <input name="inputDesencryptar">
      <soap:body use="literal"
        namespace="http://tutorial.adictosaltrabajo.com/wsd/cryptoServiceWSDL"/>
    </input>
    <output name="outputDesencryptar">
      <soap:body use="literal"
        namespace="http://tutorial.adictosaltrabajo.com/wsd/cryptoServiceWSDL"/>
    </output>
  </operation>
</binding>
<service name="cryptoServiceWSDLService">
  <port name="cryptoServiceWSDLPort" binding="tns:cryptoServiceWSDLBinding">
    <soap:address location="http://localhost:8080/cryptoService"/>
  </port>
</service>
</definitions>

```

2009-09-23
EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.

2009-09-21
Primeros pasos con JBoss Seam

2009-09-21
Integración con sistemas de Bug Tracking desde NetBeans 6.7

2009-09-14
EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.

2009-09-12
Instalación de Liferay en Tomcat existente

2009-09-11
Release Bugzilla Maven Plugin

2009-09-11
Enlazar Bugzilla con MavenChangesPlugin

2009-09-08
Sobre las reglas de codificación o... ¿de dónde salen esos caracteres "raros"?

2009-08-28
Cómo hacer deploy del site de Maven en SourceForge

2009-08-26
Ordenación por cantidades en informe cruzado

2009-08-20
Selenium IDE-Incorporando while en los test

2009-08-14
Blender y JMonkeyEngine. Exportación de archivos Blender y uso de los mismos en JMonkeyEngine

2009-08-14
5º tutorial TNT Concept Versión 0.16.1 Gestión de informes, vacaciones y utilidades

2009-08-14
Joomla 1.5. Instalación y configuración

2009-08-13
Introducción a los diagramas EPC (Event-Driven Process Chain)

2009-08-10
Blender. Animaciones avanzadas y renderización

2009-08-10
Gestión de Calidad, tablón y seguimiento en TNT Concept Versión 0.16.1

2009-08-10

Implementación con Visual Studio 2008. WSCF.blue

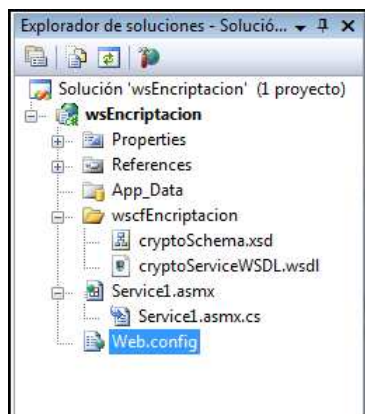
A partir del descriptor generaremos los stubs en C# del servicio que ofrezca las dos operaciones; posteriormente damos cuerpo al webservice y añadimos la lógica de negocio. En .NET podemos hacerlo a través del plugin/add-in para Visual Studio 2008, [WSCF.blue](http://msdn.microsoft.com/en-us/magazine/ee335699.aspx). En el MSDN hay un interesante artículo publicado, <http://msdn.microsoft.com/en-us/magazine/ee335699.aspx>, sobre Contract-First Web Services y WSCF.blue, e incluye el enlace <http://se.ethz.ch/~meyer/publications/computer/contract.pdf> al artículo original de [Bertrand Meyer](#): *Applying "Design by Contract"*, aparecido en el IEEE en 1992

Instalación de WSCF.blue

Descargamos el instalador del acceso Download de <http://www.codeplex.com/WSCFblue/> y lo instalamos siguiendo el asistente, de manera habitual.

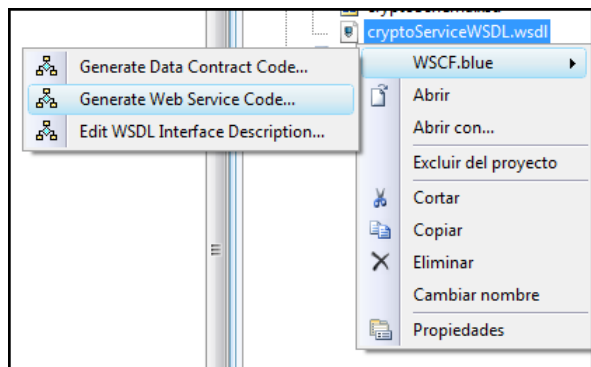
Crear el servicio e importar el contrato

En Visual Studio 2008 creamos un nuevo proyecto de tipo Web de Visual C#. Utilizaremos la plantilla: Aplicación de servicio web de ASP.NET y damos como nombre wsEncryption. Agregamos una nueva carpeta llamada wscfEncryption y a continuación nuestros WSDL y Schema:



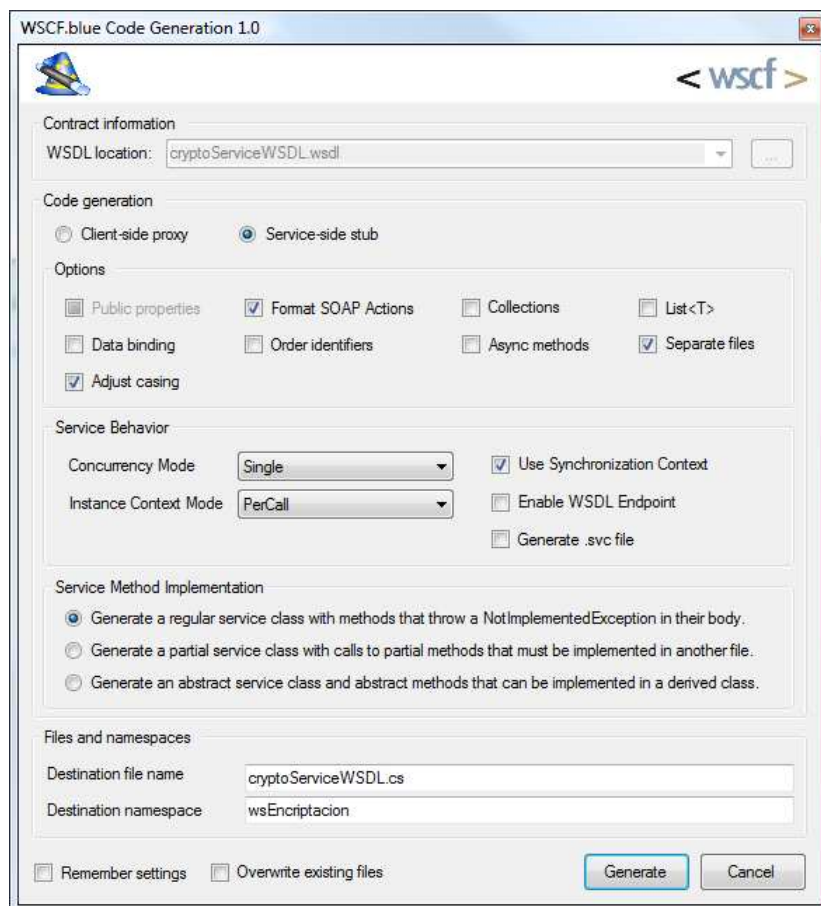
Proyecto creado en Visual Studio 2008

Bien en el menú Herramientas | Web Services Contract-First o bien pulsando con el botón derecho sobre cryptoServiceWSDL.wsdl, accedemos a las funcionalidades de WSCF.blue y escogemos Generate Web Service Code:



Menú contextual de WSCF.blue

Establecemos las opciones de generación de código siguientes:



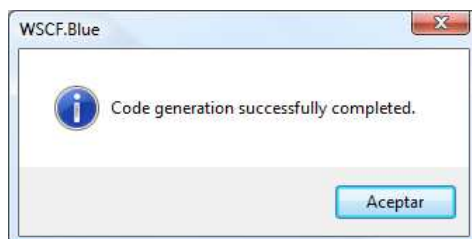
Generación de código con WSCF.blue

Como se observa en la imagen, hemos establecido las opciones siguientes:

- Server-side stub: vamos a generar el web service proveedor
- Format SOA Actions: determinamos que las acciones SOAP de cada operación del contrato sigan el formato estándar de contratos de web services: `<namespace>/<service>/<operation>[Response]`
- Separate files: cada tipo de dato del servicio web será generado en un fichero de C# distinto
- Adjust casing: los tipos de datos generados seguirán la notación de .NET
- Use synchronization context: el contexto determinará en qué hilo se ejecutará el web service
- Generate a regular service class ...: permitiremos generar métodos en donde implementar la lógica del servicio

Estas opciones son expuestas a modo de ejemplo. Cada proyecto requerirá de la configuración más adecuada para sus propósitos. La documentación se encuentra en: <http://www.thinktecture.com/resourcearchive/tools-and-software/wscf/wscf-walkthrough>

Pulsamos en **Generate** y en unos instantes se confirmará la ejecución correcta de la operación:



Código generado correctamente

En el explorador de soluciones podemos ver el código que ha sido generado:

[Cómo hacer una página web](#)

2009-08-06
[Tips And Tricks JUnit Spring](#)

2009-08-03
[Instalación de VirtualBox PUEL](#)

2009-08-03
[Gestión de contactos y pedidos en TNT Concept versión 0.16.1](#)

2009-08-03
[Comentando el libro: La estrategia del océano azul](#)

2009-07-30
[Funciones esenciales para crear un juego.](#)

2009-07-30
[2º tutorial TNT Concept versión 1.16.1](#)

2009-07-29
[Hibernate Search, Bridges, Analizadores y más](#)

Últimas ofertas de empleo

2009-07-31
[T. Información - Operador \(día / noche\) - BARCELONA.](#)

2009-06-25
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-19
[Otras - Ingeniería \(minas, puentes y puertos\) - VALENCIA.](#)

2009-06-17
[Comercial - Ventas - ALICANTE.](#)

2009-06-03
[Comercial - Ventas - VIZCAYA.](#)

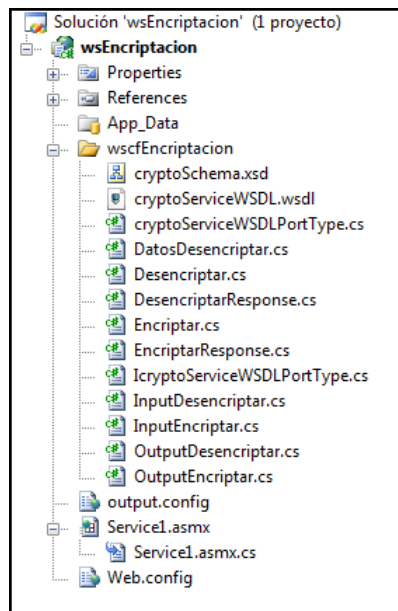
Anuncios Google

[XML Schema](#)

[Shaving Soap](#)

[Softsoap Hand Soap](#)

[Web Services](#)



Stubs generados del web service proveedor

Implementar el web service

La clase `cryptoServiceWSDLPortType` es la que finalmente alberga los métodos donde se implementará la lógica de negocio de nuestro servicio web. En su definición vemos que es una clase derivada de la interfaz `IcryptoServiceWSDLPortType`. En este tutorial implementaremos el fichero ASP web service, `Service1.asmx`, directamente con estos métodos. Para ello editamos el código de `Service1.asmx` y trasladamos el código de `cryptoServiceWSDLPortType.cs` de la siguiente manera.

- Los atributos de compilación de la clase `cryptoServiceWSDLPortType` los establecemos en la clase de `Service1`
- La clase `Service1` la convertimos en derivada de `IcryptoServiceWSDLPortType`
- Los métodos encriptar y desencriptar de `cryptoServiceWSDLPortType` los copiamos como métodos públicos no virtuales en la clase de `Service1` con el atributo `[WebService]`

Finalmente eliminamos de cada método la sentencia de lanzamiento de excepción y añadimos nuestra lógica de encriptación/desencriptación. En este caso me he basado en el post: http://www.csharper.net/blog/library_encrypt_and_decrypt_methods_using_tripleDES_and_md5.aspx. En resumen, `Service1.asmx.cs` queda de la forma:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Text;
using System.Security.Cryptography;

namespace wsEncriptacion
{
    /// <summary>
    /// Service1.asmx
    /// Tutorial: Contract-First web services con Visual Studio 2008
    /// www.adictosaltrabajo.com - Ivan Garcia Puebla
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [System.ServiceModel.ServiceBehaviorAttribute(InstanceContextMode =
        System.ServiceModel.InstanceContextMode.PerCall,
        ConcurrencyMode = System.ServiceModel.ConcurrencyMode.Single)]
    public class Service1 : System.Web.Services.WebService, IcryptoServiceWSDLPortType
    {

        [WebMethod]
        public OutputEncriptar Encriptar(InputEncriptar request)
        {

            string textoAEncriptar = request.Encriptar.Original;
            string clave = request.Encriptar.Clave;
            bool usarHashing = request.Encriptar.UsarHashing;

            byte[] keyArray;
            byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(textoAEncriptar);

            if (usarHashing)
            {
                MD5CryptoServiceProvider hashmd5 = new MD5CryptoServiceProvider();
                keyArray = hashmd5.ComputeHash(UTF8Encoding.UTF8.GetBytes(clave));
                hashmd5.Clear();
            }
            else
                keyArray = UTF8Encoding.UTF8.GetBytes(clave);

            TripleDESCryptoServiceProvider tdes = new TripleDESCryptoServiceProvider();
            tdes.Key = keyArray;
```

```
tdes.Mode = CipherMode.ECB;
tdes.Padding = PaddingMode.PKCS7;

ICryptoTransform cTransform = tdes.CreateEncryptor();
byte[] resultArray =
    cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);

string textoEncriptado = Convert.ToBase64String(resultArray, 0, resultArray.Length);

tdes.Clear();

OutputEncriptar output = new OutputEncriptar();
EncriptarResponse response = new EncriptarResponse();
response.Resultado = textoEncriptado;
output.EncriptarResponse = response;

return output;
}

[WebMethod]
public OutputDesencriptar Desencriptar(InputDesencriptar request)
{
    string textoADesencriptar = request.Desencriptar.DatosDesencriptar.Encriptado;
    string clave = request.Desencriptar.DatosDesencriptar.Clave;
    bool usarHashing = request.Desencriptar.DatosDesencriptar.UsarHashing;

    byte[] keyArray;
    byte[] toEncryptArray = Convert.FromBase64String(textoADesencriptar);

    if (usarHashing)
    {
        MD5CryptoServiceProvider hashmd5 = new MD5CryptoServiceProvider();
        keyArray = hashmd5.ComputeHash(UTF8Encoding.UTF8.GetBytes(clave));
        hashmd5.Clear();
    }
    else
        keyArray = UTF8Encoding.UTF8.GetBytes(clave);

    TripleDESCryptoServiceProvider tdes = new TripleDESCryptoServiceProvider();
    tdes.Key = keyArray;
    tdes.Mode = CipherMode.ECB;
    tdes.Padding = PaddingMode.PKCS7;

    ICryptoTransform cTransform = tdes.CreateDecryptor();
    byte[] resultArray =
        cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);

    string textoDesencriptado = UTF8Encoding.UTF8.GetString(resultArray);

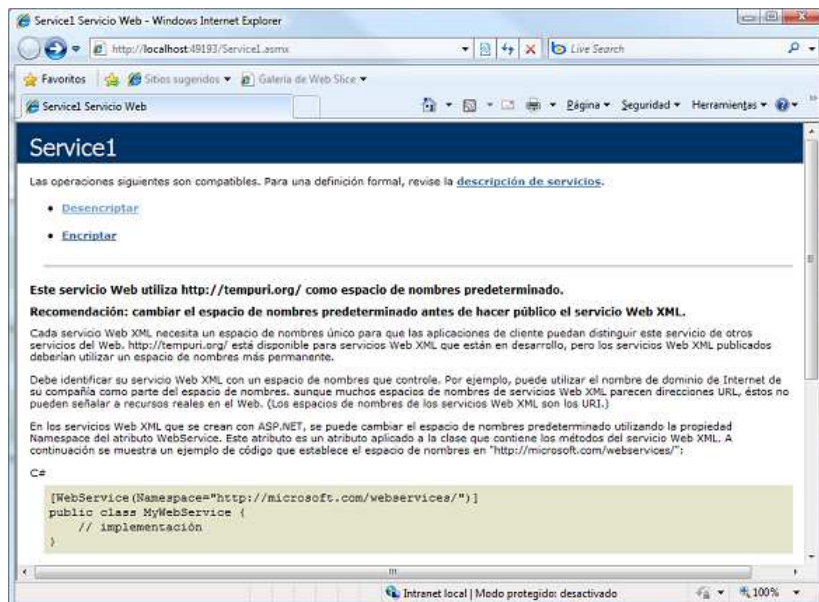
    tdes.Clear();

    OutputDesencriptar output = new OutputDesencriptar();
    DesencriptarResponse response = new DesencriptarResponse();
    response.Resultado = textoDesencriptado;
    output.DesencriptarResponse = response;

    return output;
}
}
```

Publicar y acceder al servicio

Iniciamos la depuración del proyecto y en la URL <http://localhost:49193/Service1.asmx> accedemos a la descripción del servicio web:



Probar el web service con soapUI

Para probar nuestro servicio utilizaremos soapUI (ejemplo de uso en el tutorial de [Roberto Canales: Servicio Web con NetBeans 6 y prueba con SoapUI](#)). El descriptor de nuestro servicio es `http://localhost:49193/Service1.asmx?wsdl`.

Creamos un mensaje SOAP request hacia la operación Encriptar con valores de ejemplo:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:tem="http://tempuri.org/">
  <soap:Header/>
  <soap:Body>
    <tem:Encriptar>
      <tem:request>
        <tem:Encriptar>
          <original>Lorem ipsum dolor sit</original>
          <clave>autentia2009</clave>
          <usarHashing>true</usarHashing>
        </tem:Encriptar>
      </tem:request>
    </tem:Encriptar>
  </soap:Body>
</soap:Envelope>
```

y obtenemos como mensaje SOAP response:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <EncriptarResponse xmlns="http://tempuri.org/">
      <EncriptarResult>
        <EncriptarResultResponse>
          <resultado xmlns="">m2K6Pf20MrNvX3uKR1e54KqpzLxnHmR0</resultado>
        </EncriptarResultResponse>
      </EncriptarResult>
    </EncriptarResponse>
  </soap:Body>
</soap:Envelope>
```

Realizamos la operación inversa, descriptando el resultado anterior. El mensaje SOAP request será:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:tem="http://tempuri.org/"
  xmlns:cry="http://tutorial.adictosaltrabajo.com/schema/cryptoSchema">
  <soap:Header/>
  <soap:Body>
    <tem:Desencriptar>
      <tem:request>
        <tem:Desencriptar>
          <datosDesencriptar>
            <cry:encriptado>m2K6Pf20MrNvX3uKR1e54KqpzLxnHmR0</cry:encriptado>
            <cry:clave>autentia2009</cry:clave>
            <cry:usarHashing>true</cry:usarHashing>
          </datosDesencriptar>
        </tem:Desencriptar>
      </tem:request>
    </tem:Desencriptar>
  </soap:Body>
</soap:Envelope>
```

El mensaje de respuesta obtenido concuerda con el esperado:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DesencriptarResponse xmlns="http://tempuri.org/">
      <DesencriptarResult>
        <DesencriptarResponse>
          <resultado xmlns="">Lorem ipsum dolor sit</resultado>
        </DesencriptarResponse>
      </DesencriptarResult>
    </DesencriptarResponse>
  </soap:Body>
</soap:Envelope>
```

Nuestro web service ha sido implementado correctamente partiendo de la descripción de un contrato.

Conclusión

En el [Microsoft SDK](#) existe la utilidad [ServiceModel Metadata Utility Tool \(Svcutil.exe\)](#) que permite asimismo generar las clases proxy del web service a partir del descriptor. Utilizar WSCF.blue no requiere de la instalación del SDK, no obstante, y aporta unas capacidades añadidas al propio IDE Visual Studio bastante interesantes para propósitos Contract-first.

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo Malo Regular Bueno Muy bueno



Votar

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

Autor

Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.  ADD THIS BLOG TO MY **Technorati FAVORITES**



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

[Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que

hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Creatividad Internet

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Contract-First web services con Visual Studio 2008	Utilizaremos Visual Studio 2008 con el addin WSCF.blue para generar contract-first web services (servicios web dirigidos por contrato)	2009-12-11	28	-	-	
Instalar OpenESB 2.1 e Introducción	En este tutorial veremos como descargar e instalar OpenESB y explicaremos sus funcionalidades	2009-12-03	400	-	-	
Tutorial de Google Forms	En este breve tutorial vamos a ver cómo podemos usar la herramienta Google Forms, para crear rápida y cómodamente un formulario de recogida de datos online (y de paso opinar sobre el Real Madrid).	2009-11-25	681	-	-	
Tutorial básico sobre Google Application Engine	En este tutorial veremos cómo ejecutar una simple aplicación web en la infraestructura de Google	2009-11-12	853	Bueno	1	
Axis2: Invocación de Servicios Web usando distintos MEP	En este tutorial aprenderemos a crear clientes más avanzados de servicios Web con el API de cliente de Axis2	2009-07-02	1773	-	-	
StAX (Xml Pull Parser): Streaming API para XML	En este tutorial veremos una introducción a StAX un Parser de tipo Pull y Stream XML.	2009-06-20	2309	-	-	
Registro de Web Services con Apache jUDDI. Configuración y ejemplo	Veamos como podemos catalogar y buscar web services bajo la especificación UDDI (Universal Description, Discovery and Integration)	2009-04-13	4167	-	-	
Spring WS: Servicios Web a través del correo electrónico	En este tutorial aprenderemos a configurar un cliente y servicio web para que envíe/atienda peticiones por correo electrónico.	2009-04-09	3389	-	-	
Spring WS: Construcción de Clientes de Servicios Web con Spring	En este tutorial veremos las características que nos ofrece Spring para la construcción de clientes de servicios Web	2009-03-31	2882	-	-	
Generación de XML Schema (XSD) y DTD a partir de documentos XML	En este tutorial aprenderemos a generar XML Schema (XSD) y DTD a partir de documentos XML	2009-03-15	3494	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.