CERTIFICACIÓN CFDI SAPI DE CV

CONSUMO DE WEB SERVICE

Manual de Conexión

Proceso de conexión a Web Service para diversos lenguajes de programación

Hecho en México

Revisiones

I. Control del documento

Versión	Elaboración		Revisión		Autorización	
	Nombre	**Fecha**	**Nombre**	**Fecha**	**Nombre**	**Fecha**
1.0	Gloria Martinez	28/10/16	Gustavo Acosta	28/10/16	Gustavo Acosta	28/10/16
1.1	Fernando Santiago	15/12/16	Gloria Martinez	15/12/16	Gustavo Acosta	15/12/16
1.2	Fernando Santiago	01/08/17	Gloria Martinez	01/08/17	Gustavo Acosta	01/08/17
1.3	Fernando Santiago	31/08/17	Gloria Martinez	31/08/17	Gustavo Acosta	31/08/17
1.4	Fernando Santiago	19/10/17	Gloria Martinez	19/10/17	Gustavo Acosta	19/10/17
1.5	Alfredo Albiter	15/10/18	Gloria Martinez	19/10/18	Gustavo Acosta	19/10/18

II. Histórico de revisiones

Versión	Notas
1.0	Desarrollo de Manual de Consumo de Web Service para lenguajes de programación.
1.1	Actualización de referencias, imágenes y proceso del manual de consumo.
1.2	Actualización de referencias, imágenes y proceso del manual de consumo.
1.3	Actualización de referencias, imágenes y proceso del manual de consumo. Adición de ejemplos de mensajes SOAP del proceso de certificación.
1.4	Actualización de referencia URL del servicio en el proceso del manual de consumo para versión de CFDI 3.3.
1.5	Adición de nuevos métodos para el proceso de cancelación de CFDI 3.3

Contenido

Web Service

Un Web Service es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, las cuales se comunican y pueden ser desarrolladas en lenguajes de programación diferentes, pero además pueden utilizar los servicios web para el intercambio de datos como el Internet.

Web Service describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuáles son los servicios disponibles.

Los Web Services permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información. A diferencia de los modelos Cliente/Servidor, tales como un servidor de páginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, los Web Services comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red.

Tecnología Web Services

Los Web Services están construidos con varias tecnologías que trabajan conjuntamente con los estándares que están emergiendo para asegurar la seguridad y operatividad, de modo de hacer realidad que el uso combinado de varios Web Services esté garantizado.

XML. ***E**xtensible **M**arkup **L**anguage*. El XML es una especificación desarrollada por W3C. Permite a los desarrolladores crear sus propios *tags*, que les permiten habilitar definiciones, transmisiones, validaciones, e interpretación de los datos entre aplicaciones y entre organizaciones.

SOAP. ***S**imple **O**bject **A**ccess **P**rotocol*. Es un protocolo de mensajería construido en XML que se usa para codificar información de los requerimientos de los Web Services y para responder los mensajes antes de enviarlos por la red. Los mensajes SOAP son independientes de los sistemas operativos y pueden ser transportados por los protocolos que funcionan en la Internet, como ser: SMTP, MIME y HTTP.

WSDL. ***W**eb **S**ervices **D**escription **L**anguage*. Es un lenguaje especificado en XML que se ocupa para definir los Web Service como colecciones de punto de comunicación capaces de intercambiar mensajes. El WSDL es parte integral de UDDI y parte del registro global de XML, en otras palabras, es un estándar de uso público.

UDDI. ***U**niversal **D**escription, **D**iscovery and **I**ntegration*. Es un directorio distribuido que opera en la Web que permite a las empresas publicar sus Web Services, para que otras empresas conozcan y utilicen los Web Services que publican, opera de manera análoga a las páginas amarillas.

Web Service de Producción

CERTIFICACION CFDI SAPI DE CV ofrece un servicio web en la siguiente ruta: https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx?WSDL, que contiene diferentes métodos para realizar el proceso de timbrado y cancelación de CFDI.

Los métodos se describen a continuación,

cancelaCFDi: Retorna un acuse de cancelación en un ambiente de producción.

El método recibe seis parámetros

- *String* Clave de Usuario.
- *String* Password.
- *String* RFC del emisor.
- *String* UUID del CFDI a cancelar.
- *byte [ ]* Archivo PFX.
- *String* Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

cancelaCFDiTest: Retorna un acuse de cancelación en un ambiente de pruebas.

El método recibe seis parámetros

- *String* Clave de Usuario.
- *String* Password.
- *String* RFC del emisor.
- *String* UUID del CFDI a cancelar.
- *byte [ ]* Archivo PFX.

- *String* Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato Zip.

cancelaRetencion: Retorna un acuse de cancelación de retenciones en un ambiente de producción.

El método recibe seis parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* RFC del emisor.
- *String [ ]* UUID del CFDI a cancelar.
- *byte [ ]* Archivo PFX.
- *String* Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

cancelaRetencionTest: Retorna un acuse de cancelación de retenciones en un ambiente de pruebas.

El método recibe seis parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* RFC del emisor.
- *String [ ]* UUID del CFDI a cancelar.
- *byte [ ]* Archivo PFX.
- *String* Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

consultaCFDi: Retorna el XML timbrado si el comprobante fue timbrado por el servicio.

El método recibe nueve parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* RFC del emisor.
- *String* Serie del comprobante.
- *String* Folio del comprobante.
- *String* RFC del receptor. Opcional
- *String* Fecha de emisión del comprobante. Opcional
- *String* Monto total del comprobante. Opcional
- *String* UUID del comprobante a consultar. Opcional

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

getCFDi: Retorna el comprobante junto con el TFD (Timbre Fiscal Digital) en un ambiente de producción.

El método recibe tres parámetros

- *String* Clave de Usuario.
- *String* Password.
- *byte [ ]* Archivo XML empacado en formato Zip.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato Zip.

getCFDiTest: Retorna el comprobante junto con el TFD (Timbre Fiscal Digital) en un ambiente de pruebas.

El método recibe tres parámetros

- *String* Clave de Usuario.

- *String* Password.
- *byte [ ]* Archivo XML empacado en formato Zip.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato Zip.

getCFDiTxt: Retorna el comprobante junto con el nodo Timbre Fiscal Digital en un ambiente de producción en formato texto.

El método recibe tres parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* Archivo XML.

Retorna una cadena de caracteres como respuesta.

- *String* Archivo XML.

getCFDiTxtTest: Retorna el comprobante junto con el nodo Timbre Fiscal Digital en un ambiente de pruebas en formato texto.

El método recibe tres parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* Archivo XML.

Retorna una cadena de caracteres como respuesta.

- *String* Archivo XML.

getFecha: Retorna fecha y hora del servicio en formato yyyy-MM-ddTHH:mm:ss

Retorna un valor String como respuesta.

- *String* Fecha en formato yyyy-MM-ddTHH:mm:ss.

getTimbreCFDi: Retorna únicamente el TFD (Timbre Fiscal Digital) en un ambiente de producción.

El método recibe tres parámetros

- *String* Clave de Usuario.
- *String* Password.
- *byte [ ]* Archivo XML empacado en formato ZIP.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

getTimbreCFDiTest: Retorna únicamente el TFD (Timbre Fiscal Digital) en un ambiente de pruebas.

El método recibe tres parámetros

- *String* Clave de Usuario.
- *String* Password.
- *byte [ ]* Archivo XML empacado en formato ZIP.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

getTimbreCFDiTxt: Retorna únicamente el nodo Timbre Fiscal Digital en un ambiente de producción en formato texto.

El método recibe tres parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* Archivo XML.

Retorna una cadena de caracteres como respuesta.

- *String* Archivo XML.

getTimbreCFDiTxtTest: Retorna únicamente el nodo Timbre Fiscal Digital en un ambiente de pruebas en formato texto.

El método recibe tres parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* Archivo XML.

Retorna una cadena de caracteres como respuesta.

- *String* Archivo XML.

aceptaCancelacion: Retorna acuse de aceptación del receptor para la cancelación de un CFDI

El método recibe cinco parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* UUID por aceptar la cancelación.
- *byte [ ]* Archivo PFX.
- String Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

rechazoCancelacion: Retorna acuse de rechazo del receptor para la cancelación de un CFDI

El método recibe cinco parámetros

- *String* Clave de usuario.
- *String* Password.
- *String* UUID por aceptar la cancelación.
- *byte [ ]* Archivo PFX.
- String Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

solicitaCancelacionCFDi: Retorna acuse de cancelación de CFDI cuando se requiere aceptación o rechazo del receptor.

El método recibe cinco parámetros

- *String* Clave de usuario.
- *String* Password.
- *byte []* Archivo XML que contiene el UUID que solicita la cancelación.
- *byte [ ]* Archivo PFX.
- String Contraseña del archivo PFX.

Retorna un secuencia de *byte [ ]* como respuesta.

- *byte [ ]* Archivo XML empacado en formato ZIP.

Notas:

- Los métodos “getCFDi, getCFDiTest, getTimbreCFDi, getTimbreCFDiTest” reciben un archivo XML empacado en ZIP y codificado en *base64*.
- Las respuestas satisfactorias retornan el TFD, el comprobante junto con el TFD o el acuse de cancelación en formato XML empacado en ZIP y codificado en *base64*.
- Las respuestas erróneas retornan una descripción del error generado, en formato XML empacado en ZIP y codificado en *base64*.

Estos métodos pueden ser consumidos por diversos lenguajes de programación que soporten la comunicación con un Web Service por medio de SOAP.

El presente manual describe la forma de consumir el Web Service, por los siguientes lenguajes de programación: C# .NET, VB .NET, VB6, Python, Java y PHP. Los ejemplos de Mensajes SOAP, CFDI timbrados correctamente y las respuestas de error se encuentran en la sección de Anexos.

Web Service en C# y VB .NET

Agregar Referencia Web

Para consumir un Web Service desde el lenguaje .NET (C# o VB) se debe agregar una referencia web, a continuación, se muestra el procedimiento para agregar una referencia web en el entorno de desarrollo Visual Studio.

Abrir el menú ‘Proyecto’ y seleccionar la opción ‘Agregar referencia de servicio…’.

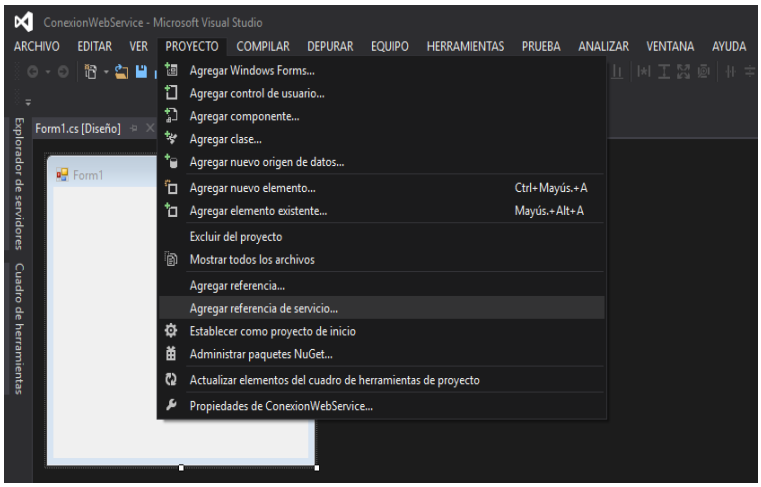

Figura 1. Opción ‘Agregar referencia de servicio’.

Se abrirá una ventana ’Agregar referencia de servicio’.

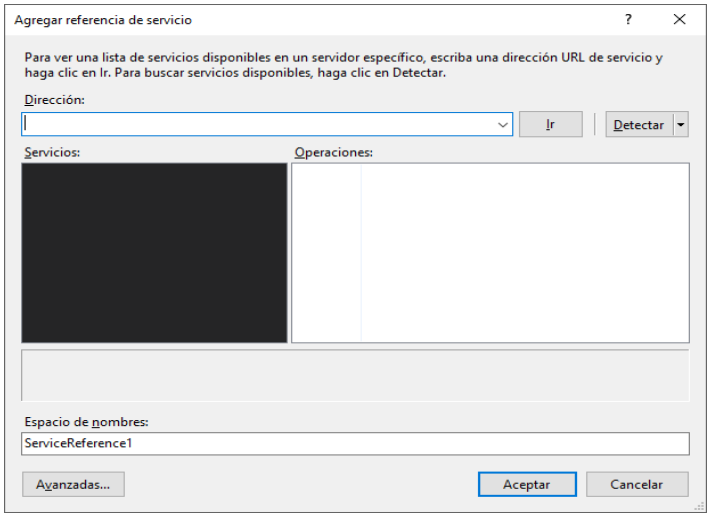

Figura 2. Ventana ‘Agregar referencia de servicio’.

Agregar en el campo ‘Dirección’ la dirección web donde se encuentra alojado el Web Service.

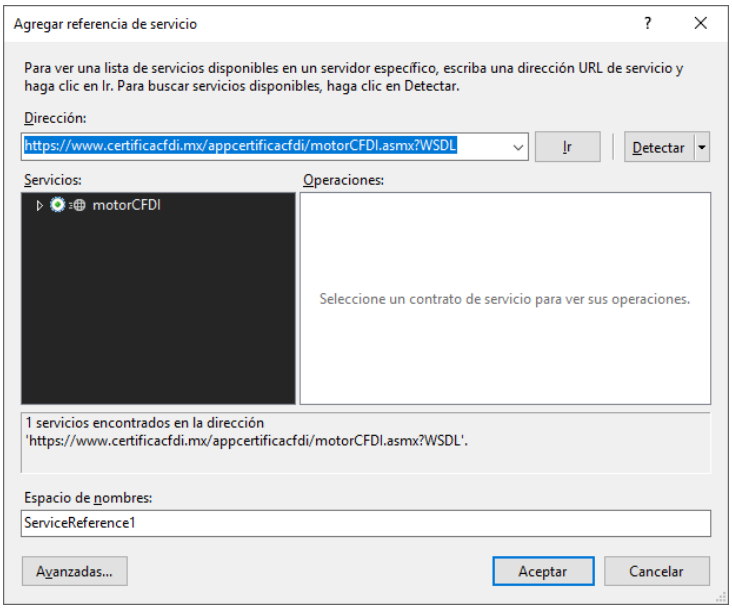

Figura 3. Conexión con el Web Service.

Una vez que se muestren los diferentes métodos del Web Service, se podrá cambiar el nombre “ServiceReference1” que se encuentra por defecto para una mejor identificación, al concluir presionar el botón ‘Aceptar’.

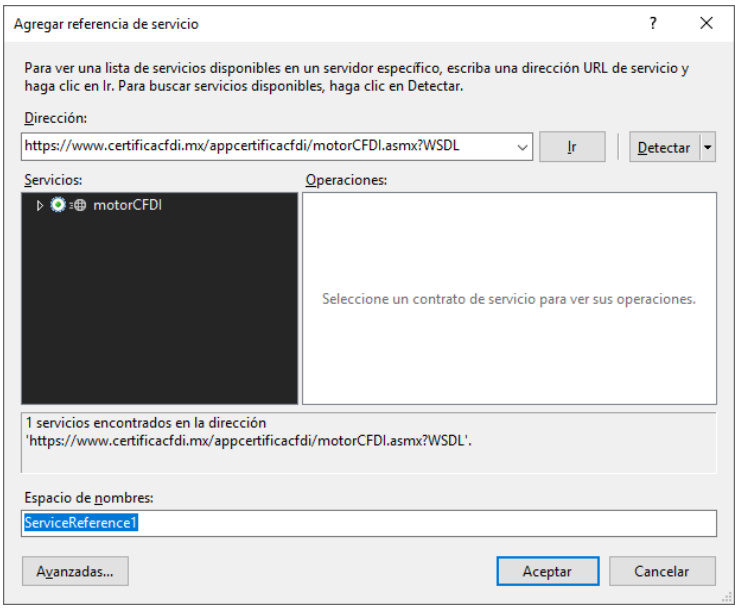

Figura 4. Agregar referencia web.

Al agregar la referencia, se generará una carpeta en el proyecto con el nombre 'Web References', donde será almacenada la referencia con el nombre aplicado.

Ejemplo: "Certificación".

NOTA: revisar el explorador de soluciones como se muestra en la Figura 5.

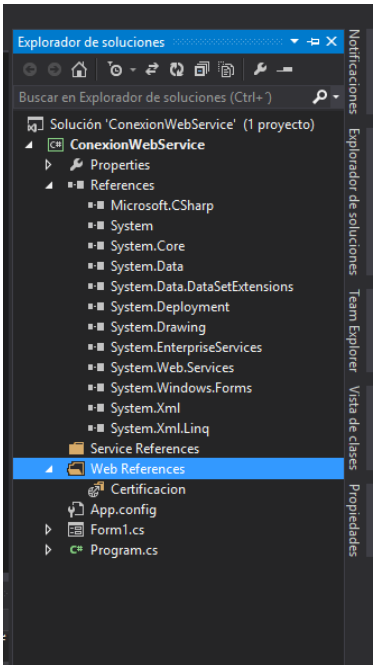

Figura 5. Localización de la referencia en el proyecto.

Configuración del cliente

En el código del proyecto se deberá realizar la importación de la referencia web con la siguiente estructura:

Ejemplo en C#

```
using Nombre del Formulario o clase seguido de un punto y el nombre de la referencia o solo el nombre de la referencia cuando se trate de Sitios Web.
```

Sintaxis de Consola o Formulario.

```
using ConexionWeb Service.Certificacion;
```

Sitio Web

```
using Certificacion;
```

Ejemplo en VB

```
Imports Nombre del Formulario o clase seguido de un punto y el
nombre de la referencia o solo el nombre de la referencia cuando
se trate de Sitios Web.
```

Sintaxis de Consola o Formulario.

```
Imports ConexionWeb Service.Certificacion
```

Sitio Web

```
Imports Certificacion
```

Una vez que se tiene la importación del espacio de nombres en el que está la clase del servicio Web, se podrá usar la clase como cualquier otra de .NET, por ejemplo, para crear un objeto de la clase *motorCFDI*, se debe realizar lo siguiente.

Ejemplo en C#

```
motorCFDI oMotor = new motorCFDI();
```

Ejemplo en VB

```
Dim oMotor As New MotorCFDI.motorCFDI
```

Con la creación del objeto se podrá acceder a los métodos del Web Service, en el siguiente ejemplo se muestra como consumir el método de *getCFDiTest*.

Ejemplo del código en C#.

```
byte[] yXML = File.ReadAllBytes(sRutaArchivoXmlEnZip);
motorCFDI oMotor = new motorCFDI();
byte[] yRespuesta = oMotor.getCFDiTest("Usuario", "password", yXML);
```

Ejemplo del código en VB.

```
Dim yXML() As Byte = File.ReadAllBytes(sRutaArchivoXmlEnZip)
Dim oMotor As New MotorCFDI.motorCFDI
Dim yRespuesta As Byte = oMotor.getCFDITest("Usuario","Password",yXML)
```

Web Service en Visual Basic 6

Agregar Referencia

Para consumir un Web Service desde el lenguaje Visual Basic 6 se debe agregar una referencia llamada ‘Microsoft Soap Type Library’ o ‘Microsoft Soap Type Library v3.0’, a continuación, se muestran los pasos a seguir.

Abrir el menú ‘Proyecto’ y seleccionar la opción ‘Referencias…’, como se muestra en la Figura 6.

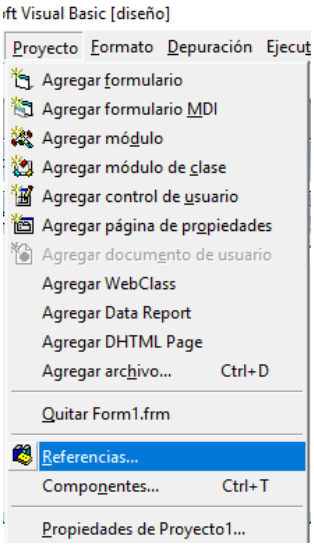

Figura 6. Opción Referencias.

Se abrirá una ventana ‘Referencias’, seleccionar de la lista de referencias ‘Microsoft Soap Type Library’ o ‘Microsoft Soap Type Library v3.0’ y presionar el botón ‘Aceptar’, como se muestra en la Figura 7.

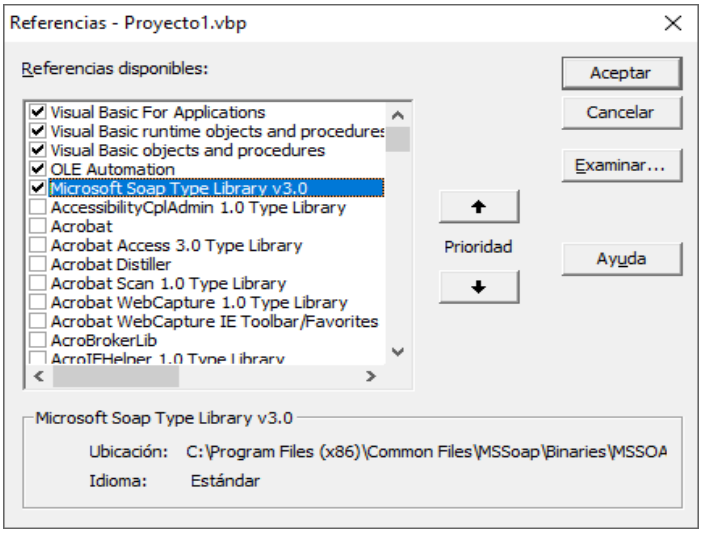

Figura 7. Agregar referencia 'Microsoft Soap Type Library'.

Configuración del cliente

Cuando este configurada la referencia 'Microsoft Soap Type Library' o 'Microsoft Soap Type Library v3.0', se podrá realizar el consumo del Web Service realizando lo siguiente:

Crear un objeto de la clase *SoapClient* o *SoapClien30*, este objeto realizará el llamado al Web Service y el consumo de los métodos que ofrezca.

```
Dim oCliente As New SoapClient30
```

Crear una etiqueta *Error* en donde se podrán obtener las excepciones o errores que puedan generarse al consumir los métodos del servicio.

```
On Error GoTo Error
Error:
MsgBox Err.Description
```

Del objeto creado, invocar el método *MSSoapInit* que recibe como parámetro un *String* con la ruta del Web Service.

```
oCliente.MSSoapInit("https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI
.asmx?WSDL")
```

De esta forma el objeto *oCliente* reconoce los métodos que se pueden utilizar del servicio y así poder invocar como cualquier otro método que tuviese el objeto de la clase *SoapClient.*

Ejemplo de consumo del método *getCFDiTest*:

```
'Declaración de variables
Dim sRutaCfdiZip As String 'Ruta del CFDI en Zip
Dim sRutaRespuesta As String 'Ruta de la respuesta del servicio
Dim ayXML() As Byte
Dim NumArchivo As Integer
Dim ayRes() As Byte
Dim fLen As Long
'Lectura del archivo a enviar
'Abrir el archivo en modo binario de solo lectura (Binary Lock Read)
Open sRutaCfdiZip For Binary Lock Read As 1
'Redimensionar el array al tamaño del archivo
fLen = FileLen(sRutaCfdiZip)
ReDim ayXML(fLen) As Byte
'Leer el archivo entero y almacenarlo en el array
Get #1, , ayXML
Close
'Consumo de Servicio Web
Dim oCliente As New SoapClient30
On Error GoTo Error
Dim sUrlWS As String
sUrlWS =""
oCliente.MSSoapInit(sUrlWS)
ayRes = oCliente.getCFDiTest("Usuario", "Contraseña", ayXML)
'Escribir la respuesta del servicio
'Crear un archivo para guardar los datos recibidos (Binary Access
Write)
Open sRutaRespuesta For Binary Access Write As 1
'Escribir los bytes del array de la respuesta, en el nuevo archivo
Put #1, , ayRes
Close
```

Web Service en Python

URL del servicio

URL: https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx?WSDL

Configuración del cliente

Configuración del cliente a través de la API de SUDS para el manejo de servicios web basados en SOAP.

La instrucción *from suds.client import Cliente* importa el módulo para ver los mensajes SOAP de entrada y salida así como las cabeceras HTTP.

Instancia al servicio web a través de la clase *Client*, recibe como parámetro la URL del servicio.

```
wsURL = ""
oCliente = Client(wsURL)
```

Llamado a los métodos del servicio a través del objeto *oCliente*.

```
respuesta = oCliente.service.getCFDiTest(string, string, base64)
```

Ejemplo

```
import zipfile              #Lectura y Escritura de archivos Zip
import os                   #Manejo de archivos y directorios
from suds.client import Client       #Configuración del cliente
import base64                        #Conversión a base64

#Ruta del XML original
xml = os.path.relpath("AAA010101AAA1.xml")

#Ruta del Zip generado a partir del XML original
xmlZip = os.path.relpath("AAA010101AAA1.zip")

#Ruta del Zip de respuesta
xmlResp = os.path.relpath("Respuesta.zip")

#Generación del archivo Zip a partir del XML original
zpFile = zipfile.ZipFile(xmlZip, "w")
```

```
zpFile.write(xml)
zpFile.close()

#Recupera el archivo zip
oZip = open(xmlZip, "rb")
yXml = oZip.read()
oZip.close()

#Configuración del cliente
wsURL = ""
oCliente = Client(wsURL)

#Convierte a base64 antes de enviar el Zip
paquete = base64.b64encode(yXml).decode("utf-8")

#Llamado al método getCFDiTest() y recibe la respuesta del servicio
respuesta = oCliente.service.getCFDiTest(["usuario"], ["password"],
paquete)

#Descodifica base64 la respuesta del servicio
yResp = base64.b64decode(respuesta)

#Crea el Zip de respuesta
oResp = open(xmlResp, "wb")
oResp.write(yResp)
oResp.close()

#Extrae el XML dentro del Zip de respuesta
unZip = zipfile.ZipFile(xmlResp)
unZip.extractall()
unZip.close()
```

Web Service en Java

Agregar Referencia

Para consumir un Web Service desde el lenguaje Java se debe agregar una referencia llamada 'Web Service Client…', a continuación, se muestra el procedimiento para agregar una referencia en el entorno de desarrollo *NetBeans IDE 8.2 english version.*

Dar *clic* derecho en el proyecto creado y seleccionar la opción 'New', se expandirá un submenú, donde se deberá seleccionar la opción 'Other…', como se muestra en la Figura 8.

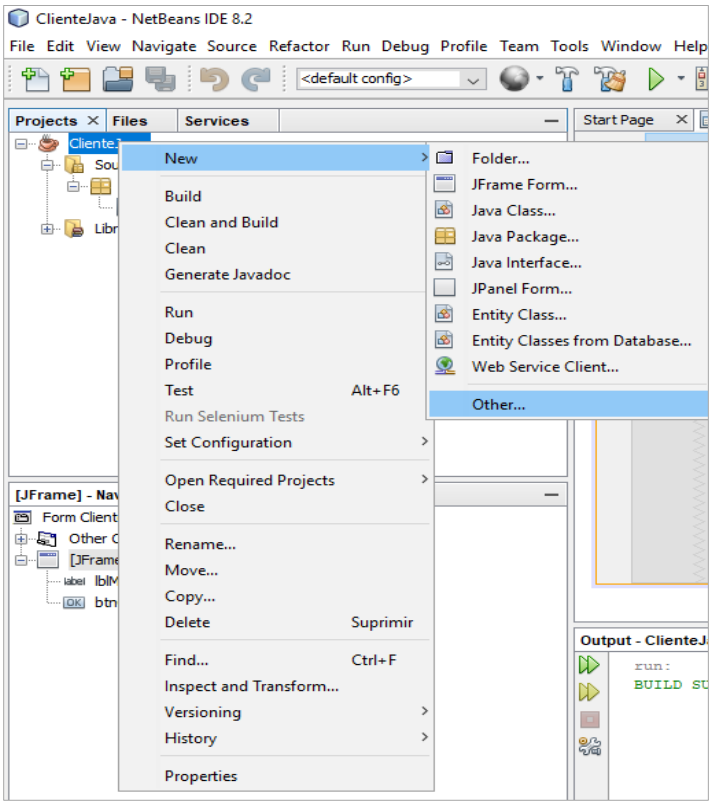

Figura 8. Opción 'Other…'.

Se abrirá una ventana 'New File', seleccionar de la lista 'Categories' la opción 'Web Services', se actualizará la lista 'File Types' en donde se deberá seleccionar la opción 'Web Service Client' y presionar el botón 'Next >', como se muestra en la Figura 9.

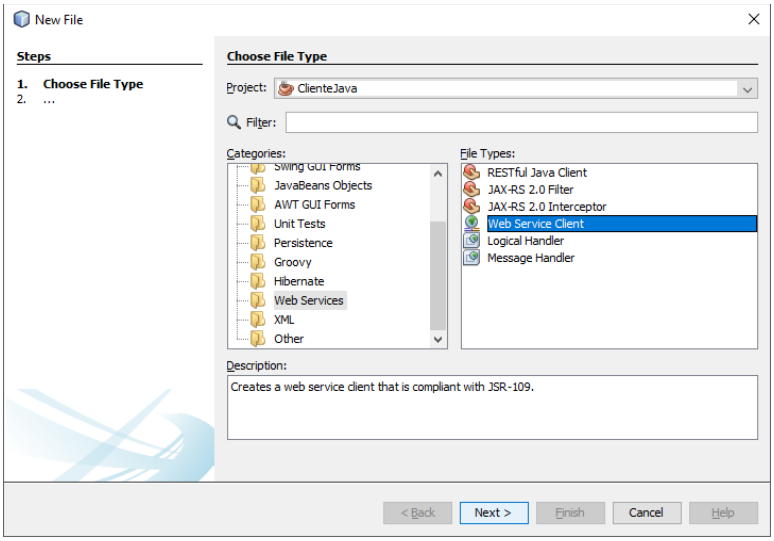

Figura 9.Categoria 'Web Service Client'.

Se abrirá una ventana 'New Web Service Client', seleccionar la opción 'WSDL URL', escribir la ruta donde se encuentra el Web Service, como se muestra en la Figura 10.

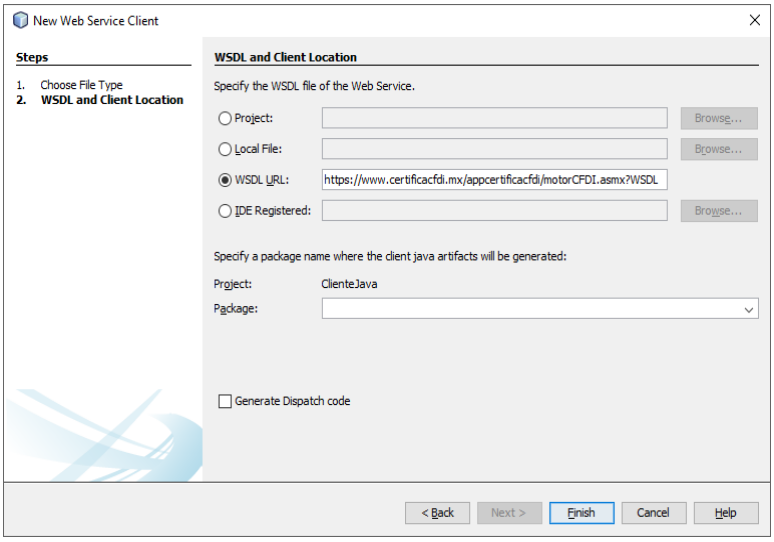

Figura 10. Ruta del Web Service.

Una vez escrita la ruta del Web Service, en el campo 'Package' escribir un nombre para identificar el paquete que se generará al finalizar, por ejemplo 'Cliente.Servicio' y presionar el botón 'Finish', como se muestra en la Figura 11.

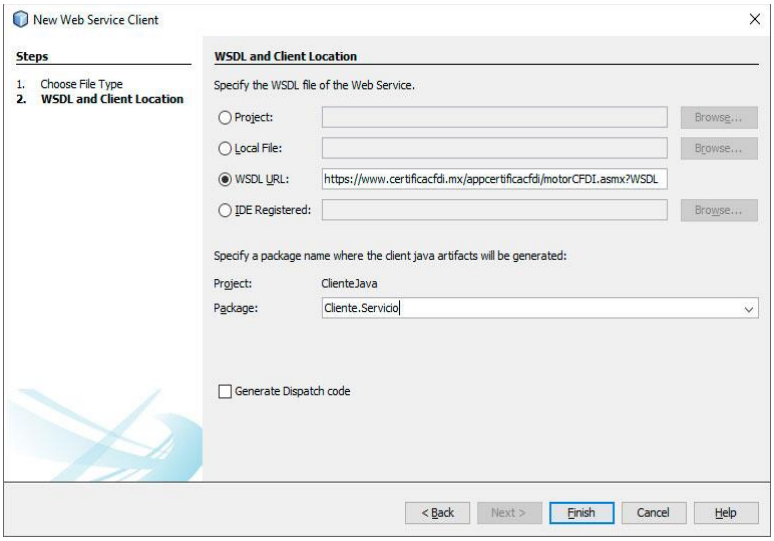

Figura 11. Adicionar nombre a 'Package'.

Al agregar la referencia, se generará una carpeta en el proyecto con el nombre "Generated Sources (jax-ws)", donde será almacenado el *Package*, también se generará otra carpeta con el nombre "Web Service References", donde será almacenada la referencia del Web Service. NOTA: revisar el explorador como se muestra en la Figura 12.

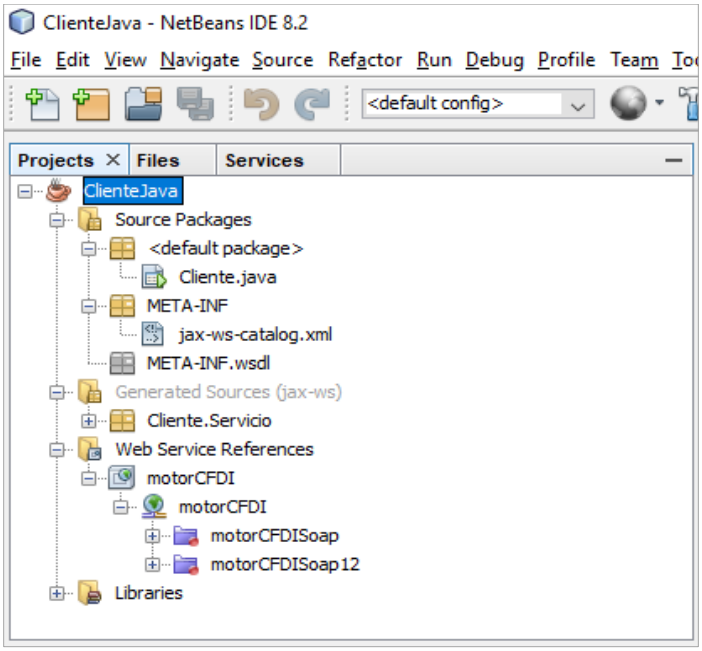

Figura 12. Explorador de proyectos.

Configuración del cliente

Cuando este configurada la referencia, se podrá realizar el consumo del Web Service realizando lo siguiente.

Importar los espacios de nombres *MotorCFDI* y *MorotCFDISoap* para consumir los métodos del Web Service.

```
import Cliente.Servicio.MotorCFDI;
import Cliente.Servicio.MotorCFDISoap;
```

Crear un objeto de la clase *MotorCFDI* e instanciarlo.

```
MotorCFDI oServicio = new MotorCFDI();
```

Crear un objeto de la clase *MotorCFDISoap* que será igual el método *getMotorCFDISoap* del objeto de la clase *MotorCFDI.*

```
MotorCFDISoap oMotor = oServicio.getMotorCFDISoap();
```

De esta forma el objeto *oMotor* reconoce los métodos que se pueden utilizar del servicio, en el siguiente ejemplo muestra como consumir el método de *getCFDiTest*.

Ejemplo del Cliente en Java

```
//Importaciones necesarias
import Cliente.Servicio.MotorCFDI;
import Cliente.Servicio.MotorCFDISoap;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

//Importaciones opcionales
import javax.swing.JOptionPane;


//Código de consumo del servicio
try
  {
```

```
    //Rutas de archivos
    String sRutaCFDIZip;      //Ruta del CFDI empacado en ZIP
    String sRutaRespuesta;    //Ruta de la respuesta del servicio

    //Credenciales
    String sUsuario="Usuario";
    String sPassword="Contraseña";

    //Lectura del CFDI en Zip
    File oFichero = new File(sRutaCFDIZip);
    byte aCFDI[] = new byte[(int)oFichero.length()];
    try(FileInputStream fisFichero = new FileInputStream(oFichero))
    {
        fisFichero.read(aCFDI);
    }
    //Fin de la Lectura

    //Consumo de Web Service
    MotorCFDI oServicio = new MotorCFDI();
    MotorCFDISoap oMotor = oServicio.getMotorCFDISoap();
    byte aRespuesta[];
    aRespuesta=oMotor.getCFDiTest(sUsuario, sPassword, aCFDI);
    //Fin de Consumo


    //Guardar respuesta
    File oRes = new File(sRutaRespuesta);
    try(FileOutputStream fosFichero = new FileOutputStream(oRes))
    {
        fosFichero.write(aRespuesta);
    }
    //Fin de Guardar respuesta
    JOptionPane.showMessageDialog(null, "Envío con éxito");
}
catch(IOException ex)
{
    JOptionPane.showMessageDialog(null, ex.getMessage());
}
```

Web Service en PHP

Configuración del cliente

Para poder consumir un Web Service desde el lenguaje PHP, se requiere de la clase *SoapClient*, esta clase recibe como parámetro un *String* con la ruta del Web Service.

```
$sUrlWS='https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx?WSDL ';
$oCliente = new SoapClient($sUrlWS);
```

Al consumir cualquier método que ofrece el Web Service, se requiere crear una variable que contenga un *array (*arreglo) con los parámetros que se definan en el método, por ejemplo:

```
$aParametros = array('psUsuario'=>$sUsuario,
                     'psPassword'=>$sPassword,
                     'pyXML'=>$contenido);
```

Nota: Los parámetros deben ser igual a los que están definidos por el método del servicio para que no cause un conflicto al momento de consumir el método.

El llamado al método se debe realizar de la siguiente manera:
Crear una variable que contenga la respuesta del método, por ejemplo.

```
$aRespuesta = $oCliente->getCFDiTest($aParametros)->getCFDiTestResult;
```

Con lo anterior se pueden consumir los métodos del servicio, a continuación, se muestra un ejemplo del cliente en PHP.

```
$sRutaCFDI;       //Ruta del CFDI a empacar
$sNomZip;         //Ruta del CFDI empacado
$sRutaZipRes;     //Ruta donde se guardará el ZIP de respuesta
$sRutaRespuesta;  //Ruta para desempacar la respuesta

//Empacar CFDI en ZIP
//Objeto de la clase ZipArchive
$oZip = new ZipArchive();

if($oZip->open($sNomZip,ZIPARCHIVE::CREATE)===true)
```

```
{
	$oZip->addFile($sRutaCFDI);
	$oZip->close();
}
//Fin del proceso de empacado

//Lectura en bytes del CFDI empacado
$gestor = fopen($sNomZip, "r");
$contenido = fread($gestor, filesize($sNomZip));
fclose($gestor);
//Fin de la lectura

// Llamada al Web Service
// Creación del objeto oCliente de la clase SoapClient
$sUrlWS='https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx?WSDL;
$oCliente = new SoapClient($sUrlWS);

//Se crean dos variables que contienen las credenciales para acceder al servicio
$sUsuario='Usuario';
$sPassword='Password';

//Se crea una variable $aParametros que contiene un array que hace referencia a los parámetros definidos en el método del Web Service
$aParametros = array('psUsuario'=>$sUsuario,
                     'psPassword'=>$sPassword,
                     'pyXML'=>$contenido);
//Se connsume el método getCFDiTest para recibir su respuesta "getCFDiTestResult"
$aRespuesta=$oCliente->getCFDiTest($aParametros)->getCFDiTestResult;
//Fin del llamado al método

//Se guarda la respuesta obtenida
$oZipRes = fopen($sRutaZipRes,'w');
fwrite($oZipRes,$aRespuesta);
fclose($oZipRes);

//Se crea un objeto ZIP de la clase ZipArchive
$oZipR = new ZipArchive();



//Se abre el ZIP para extraer los datos de la respuesta del servicio
$bRes = $oZipR->open($sRutaZipRes);

//Se extrae el contenido del Zip
$oZipR->extractTo($sRutaRespuesta);
$oZipR->close();
```

Anexos

Petición SOAP

En el siguiente cuadro se muestra una Petición SOAP que se realiza al Web Service de CERTIFICACIÓN CFDI.

```
POST https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client
Protocol 4.0.30319.42000)
VsDebuggerCausalityData:
uIDPo3VsMEgFBwRDnt90RUpVFP8AAAAABiNB9kN+5ESc+vDTGv1h7U8aP1Gpwn5OlAxHK4qxqPsAC
QAA
Content-Type: text/xml; charset=utf-8
SOAPAction:
"https://www.certificacfdi.mx/appcertificacfdi33/motorCFDI.asmx?WSDL/getCFDi"
Host: www.certificacfdi.mx
Content-Length: 3892
Expect: 100-continue
Connection: Keep-Alive

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
      <getCFDi xmlns="">
            <psUsuario>Usuario</psUsuario>
            <psPassword>Password</psPassword>
            <pyXML>H4sIAAAAAAAAEANVWyZKjSBI9z5jNP8h0rakUILSlVVZbs
            ApEgFiFuIyxCSGxCZBA/EF/Ux/m0B80vzCBlIuyuqxnLnOYTEOA4
            +H+nrtHuP/rt39++6VNk</pyXML>
      </getCFDi>
      </soap:Body>
</soap:Envelope>
```

Respuesta SOAP

En el siguiente cuadro se muestra una Respuesta Soap que realiza al Web Service de CERTIFICACIÓN CFDI.

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 30 Aug 2017 16:57:18 GMT
Content-Length: 4762

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <soap:Body>
            <getCFDiResponse
             xmlns="">
      <getCFDiResult>H4sIAAAAAAAEANVWyZKjSBI9z5jNP8h0rakUIL
                  SlVVZbsApEgFiFuIyxCSGxCZBA/EF/Ux/m0B80vzCBlIuyuqxnLnO
                  YTEOA4+H+nrtHuP/rt39+6VNk</getCFDiResult>
            </getCFDiResponse>
      </soap:Body>
</soap:Envelope>
```

CFDI Correcto

Ejemplo de un CFDI timbrado correctamente por el Web Service de CERTIFICACIÓN CFDI.

<?xml version="1.0" encoding="UTF-8"?>
<cfdi:Comprobante xmlns:cfdi="http://www.sat.gob.mx/cfd/3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sat.gob.mx/cfd/3
http://www.sat.gob.mx/sitio_internet/cfd/3/cfdv33.xsd" Version="3.3" Serie="A" Folio="00063"
Fecha="2017-12-04T18:57:50" FormaPago="01" CondicionesDePago="Contado"
NoCertificado="20001000000300022755"
Certificado="MIIF7TCCA9WgAwIBAgIUMjAwMDEwMDAwMDAzMDAwMjI3NTUwDQYJKo
Ho6TR1nduAsJ8s89MZ9P2D9OvbuKkkSwbsXzL02F0udKdkN1/XkKWsWOBS/9WxC+cn"
Sello="BXvn/6ydA8By+2rM06hFG9zEz3rOdzZzLsu8xTyOWtwCvwRzg+uwWu8VoZIwAOf/
msW81Q2Pe0yBcO8DacFtTlwjpK7hs/RKB8w==" SubTotal="15.00" Descuento="5.00"
Moneda="MXN" TipoCambio="1" Total="11.60" TipoDeComprobante="I" MetodoPago="PUE"
LugarExpedicion="55067">
  <cfdi:Emisor Rfc="AAA010101AAA" Nombre="EMPRESA PRUEBAS SA DE CV"
RegimenFiscal="601" />
  <cfdi:Receptor Rfc="XAXX010101000" Nombre="RAZÓN SOCIAL" UsoCFDI="P01" />
  <cfdi:Conceptos>

```
   <cfdi:Concepto ClaveProdServ="50211503" NoIdentificacion="32A1SD3A1D" Cantidad="1.00"
ClaveUnidad="H87" Descripcion="TABACO" ValorUnitario="15.00" Importe="15.00"
Descuento="5.00">
     <cfdi:Impuestos>
      <cfdi:Traslados>
       <cfdi:Traslado Base="10.00" Impuesto="002" TipoFactor="Tasa" TasaOCuota="0.160000"
Importe="1.60" />
      </cfdi:Traslados>
     </cfdi:Impuestos>
   </cfdi:Concepto>
  </cfdi:Conceptos>
  <cfdi:Impuestos TotalImpuestosTrasladados="1.60">
   <cfdi:Traslados>
    <cfdi:Traslado Impuesto="002" TipoFactor="Tasa" TasaOCuota="0.160000" Importe="1.60" />
   </cfdi:Traslados>
  </cfdi:Impuestos>
  <cfdi:Complemento>
   <tfd:TimbreFiscalDigital xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sat.gob.mx/TimbreFiscalDigital
http://www.sat.gob.mx/sitio_internet/cfd/TimbreFiscalDigital/TimbreFiscalDigitalv11.xsd"
Version="1.1" RfcProvCertif="CCF1011111K9"
SelloSAT="P6Pphwh2AJRF4RUVmLnn827uMKrXvXBDc87z2kPUtzAal8ZSgXHPsSvVMv
/h5IH66ClqfljupNRTXnOz9zOGyq8oPLoIm4Evg==" NoCertificadoSAT="20001000000300022323"
FechaTimbrado="2017-12-04T18:58:12" UUID="B9C918F1-ED87-4A2D-9B74-1547FCC5A230"
SelloCFD="BXvn/6ydA8By+2rM06hFG9zEz3rOdzZzLsu8xTyOWtwCvwRzg+uwWu8VoZIwAmsW8
1Q2Pe0yBcO8DacFtTlwjpK7hs/RKB8w==" xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital"
/>
  </cfdi:Complemento>
</cfdi:Comprobante>
```

Respuesta de error

Las respuestas de error contienen detalles acerca de las causas del error durante el flujo de la petición al ser recibida por el Web Service de CERTIFICACIÓN CFDI, el mensaje de error es devuelto en formato XML comprimido en ZIP y codificado en *base64*.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelop xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
       <soap:Body>
              <soap:Fault>
                     <faultcode titulo="soap:Server" />
                     <faultstring mensaje="XXXX XXXXXXXXXXX" />
                     <Detail>
                            <Error codigo="XXXXX" mensaje="XXXXX XXXXX XXXXX." />
                     </Detail>
              </soap:Fault>
       </soap:Body>
</soap:Envelop>
```

Contacto

Si cuenta con alguna duda, sugerencia o bien requiere soporte, a continuación, le presentamos al personal que puede asesorarlo.

DIRECTORIO		
Tel. (55) 59.32.48.22 | 59.38.33.68 | 59.38.33.71 | 59.38.49.17		
Área	**Nombre**	**Correo**
Soporte	Ing. Gustavo Acosta Ing. Annaís Martínez	gacostab@certificacfdi.com amartinez@certificacfdi.com soporte@certificacfdi.com
Soporte de Conexión al servicio	Ing. Alfredo Albiter Ing. Fernando Santiago	albiterrojo@certificacfdi.com santiagof@certificacfdi.com
Administración	Leticia Ávila	lavila@certificacfdi.com

Si nos contacta por correo electrónico por favor enviar los datos que se mencionan en el siguiente formulario.

INFORMACIÓN DE LA SOLICITUD DE SOPORTE	
Nombre:	
Empresa:	
Teléfono:	
Correo Electrónico:	
Contacto Skype:	
Asunto:	