# Guía Técnica para la implementación de Cliente de Web Service en Visual Basic

## INDICE:

Contenido

[Guía Técnica para la implementación de Cliente de Web Service en Visual Basic 1](#_Toc481763660)

[INDICE: 2](#_Toc481763661)

[WEB SERVICE: 3](#_Toc481763662)

[Método para Crear un XML mediante Texto: 4](#_Toc481763663)

[Método para Timbrar una Nomina mediante Texto: 4](#_Toc481763664)

[Método para Sellar: 5](#_Toc481763665)

[Método para Timbrar: 6](#_Toc481763666)

[Método para Cancelar CFDI: 7](#_Toc481763667)

[Método para Recuperar Acuse: 8](#_Toc481763668)

[Método para Recuperar Acuse de Cancelación: 9](#_Toc481763669)

[Método para Obtener la fecha y hora actual del servidor: 10](#_Toc481763670)

[Implementación en Visual Basic: 11](#_Toc481763671)

[Programas Especiales Para la cancelación: 13](#_Toc481763672)

## WEB SERVICE:

Obtenga los servicios contratados, aprobados y disponibles para el usuario. Éste es un servicio web con el cuál usted podrá atreves de su propio cliente:

* Crear un XML mediante texto
* Sellar un CFDI.
* Timbrar un CFDI.
* Cancelar un CFDI.
* Recuperar Acuse.
* Recuperar Acuse cancelado.
* Obtener la fecha y hora actual del servidor.

La ruta del wsdl para consumo del web service:

https://sistema.facturaselectronicascfdi.com/wsTimbrado.asmx

Para fecha y hora la ruta del wsdl es:

https://sistema.facturaselectronicascfdi.com/wsfechahora.asmx

## Método para Crear un XML mediante Texto:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbradoTexto.asmx

Para Crear se usara el método "TextoAXml".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

String RFC.- rfc que utilizamos.

String contenidoArchivo.- contenido del archivo de texto.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

XML().- Devuelve el XML que convirtió mediante el texto.

## Método para Timbrar una Nómina mediante Texto:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbradoTexto.asmx

Para timbrar se usará el método " timbradoPorTextoNomina".

Parámetros:

String correo.- Correo electrónico registrado en nuestra página.

String pass.- Contraseña que utiliza el usuario para entrar a nuestra página

String rfc.- rfc que utilizamos.

byte() cerfile.- Archivo con extensión .cer.

byte() KeyFile.- Archivo con extensión .key del certificado codificado a base 64.

String passCer.- Contraseña privada del archivo .key.

byte() CerFile.- Archivo con extensión .cer.

byte() contenidoArchivo.- Archivo con extensión .txt que contiene la nómina a timbrar.

Boolean pdf.- Un booleano que se envía, si se quiere archivo pdf o no.

String ambiente.- Variable mediante la cual puede o no consumir folios, (“pruebas”,”produccion”).

String version.- Variable mediante la cual decidimos que versión queremos timbrar, la primera parte es la versión del comprobante seguido de un guion, en la segunda parte es la versión de la nómina, al momento solo existe la versión “3.2-1.2”, la versión “3.3-1.2” aun esta en desarrollo

## Método para Sellar:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbrado.asmx

Para sellar se usará el método "sellar".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

byte() KeyFile.- Archivo con extensión .key del certificado codificado a base 64.

String PassCsd.- Contraseña privada del archivo .key.

byte() CerFile.- Archivo con extensión .cer.

byte() xml.- Archivo con extensión XML que contiene el comprobante a sellar.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

Sello().- Un String con el sello del CFDI.

XmlSellado().- Devuelve el XML sellado.

## Método para Timbrar:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbradoTexto.asmx

Se usará el método "timbrar".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

byte() xml.- Archivo con extensión XML que contiene el comprobante a timbrar, debe estar previamente sellado.

Boolean pdf.- Un booleano que se envía, si se quiere archivo pdf o no.

String ambiente.- Variable mediante la cual puede o no consumir folios, (“pruebas”,”produccion”)

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

FolioUUID().- Un String con el folio UUID.

XML().- Devuelve el XML timbrado.

Acuse().- Devuelve el Acuse.

PDF().- Devuelve el PDF del XML si se envió el requerimiento.

## Método para Cancelar CFDI:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbrado.asmx

Se usará el método "cancelar".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

String UUID.- Un String que contiene el o los folios UUID para cancelar el CFDI.

String KeyEncode.- Encode en base 64 de la llave privada del CSD del emisor.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

Acuse().- Devuelve el Acuse.

## Método para Recuperar Acuse:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbrado.asmx

Se usará el método " recuperarAcuse".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

String UUID.- Un String que contiene el folio UUID para cancelar el CFDI.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

Acuse().- Devuelve el Acuse.

## Método para Recuperar Acuse de Cancelación:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsTimbrado.asmx

Se usará el método " RecuperarAcuseCancelacion".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

String UUID.- Un String que contiene el folio UUID para cancelar el CFDI.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

Acuse().- Devuelve el Acuse.

## Método para Obtener la fecha y hora actual del servidor:

Wsdl: https://sistema.facturaselectronicascfdi.com/wsfechahora.asmx

Se usará el método " RecuperarFechaHoraActual".

Parámetros:

String curCorreo.- Correo electrónico registrado en nuestra página.

String password.- Contraseña que utiliza el usuario para entrar a nuestra página.

La llamada al método le devolverá el objeto “Respuesta” con los siguientes métodos:

Errores().- Un String, si hubo errores, aquí se obtendrán, de lo contrario será “”.

FechaActual().- Devuelve la fecha actual del servidor.

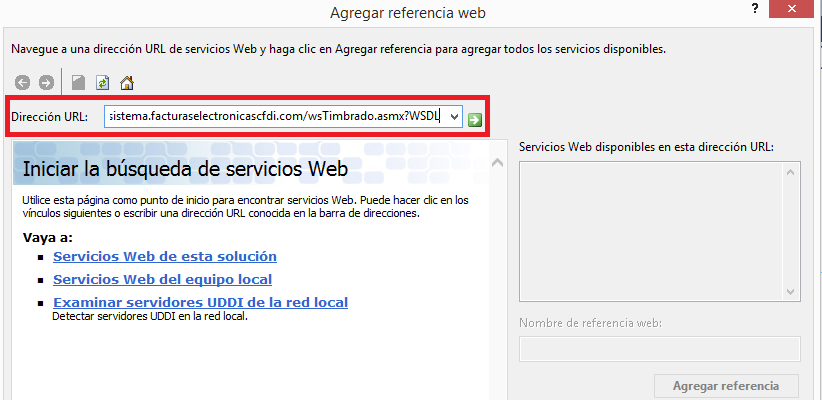
HoraActual().- Devuelve la hora actual del servidor.

FechaHoraActual().- Devuelve la hora y fecha actual del servidor.

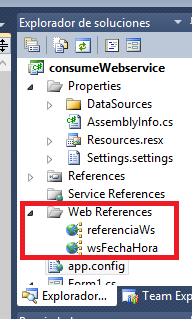
## Implementación en Visual Basic:

Para la implementación se incluye un comprimido “ClienteWebServiceVBEjemplo.zip” con un ejemplo de una aplicación en C# que realiza el consumo del Web Service, se incluyen archivos de ejemplo adjuntos.

Para esto se tiene que agregar una Referencia WEB en el programa Visual Studio de la siguiente forma, a nuestro proyecto: agregar una referencia web -> agregar referencia de servicio-> avanzadas-> agregar referencia web->”Colocar la URL del WSD” que se encuentra en nuestra página, ejemplo:



Con esto se podrá manejar los métodos y respuestas de nuestro Web Service.



## Programas Especiales Para la cancelación:

Para el caso de cancelación para codificar el archivo .key de su certificado a base 64 se tiene que generar el archivo .pem para esto se utiliza Openssl que se incluye en “ClienteWebServiceVBejemplo.zip”.

Instalación OpenSSL:

 OpenSSL v1.0.0g Light (Se incluye en el comprimido adjunto.)

 [Visual VB 2008 Redistributables](http://www.microsoft.com/en-us/download/details.aspx?id=29) (se puede descargar del link).

Instalamos Visual Basic 2008 Redistributables, seguidamente OpenSSL v1.0.0g Light.

Y se ejecuta el comando desde el código como se pude ver en el ejemplo que se incluye adjunto “pkcs8 -inform DER -in C:\ruta\aaa010101aaa\_\_csd\_01.key -passin pass:contraseña\_usuario -out C:\ruta\_destino\aaa010101aaa\_\_csd\_01.pem”.