

PLATAFORMA PARA GENERACIÓN DE CFDI

Manual Técnico



CONTENIDO

Plataforma para la generación de CFDi	4
Web Service	6
Métodos	7
FactorumGenYaSellado	7
FactorumGenYaSelladoTest	8
Ejemplos para FactorumGenYaSellado y FactorumGenYaSelladoTEST	8
FactorumGenYaSelladoConArchivo	9
FactorumGenYaSelladoConArchivoTest	10
Respuesta de los métodos	11
Validaciones que contiene el servicio	12
Lista de códigos de error	12
WebService PDF	14
Método	15
Lista de códigos de error	18
Conector Local.....	19
Explicación del archivo de configuración:	20
Explicación de las carpetas	20
Archivos de Entrada:	20
Archvos PDF.....	21
Archvos Salida	21
Archvos ERROR	21
Explicación del Layout:	22
Explicación de los Archivos 0.dat, 1.dat, etc.....	22
Explicación del flujo de información.	22
Web Conector	24
Métodos	25
Explicación del Layout:	25
Lista de códigos de error	28
WebService Portal.....	29
FAQ	32
¿Qué es un web service?.....	32
¿Para qué me sirve el WDSL?	32
¿Qué significa estar en SOAP?.....	32
¿Por qué usar XML?	32
¿Qué es el Timbre en mi factura?	32

¿Qué información hay en el código bidimensional?	32
Que es el UUID regresado por el SAT?	32
¿Qué es el sello CFD?	33
¿Qué es la cadena Original?	33
¿Qué es el no. De certificado SAT?	33
¿Qué es el sello SAT?	33
¿Quién puede usar el web service de Factorum?	33
¿En qué tecnología puedo usar el web service?	33
¿La generación del timbre mediante el consumo del web service es seguro?	33
¿Cuánto tiempo tarda en responder el web service?	33
¿Cómo puedo usarlo para probarlo y ver si me sirve?	33
¿Cómo integro el web service en mi sistema o programación?	34
¿Qué ventajas tengo al contratar el web service?	34

PLATAFORMA PARA LA GENERACIÓN DE CFDI

Esta plataforma acredita a Factorum en Proveedor de Servicios de Expedición de Comprobante Fiscal Digital a través de Internet, conforme a los nuevos requerimientos especificados para realiza todas las validaciones de los CFDi exigidas por el SAT.

La plataforma cuenta con diversos productos, de esta manera el cliente puede seleccionar aquel que cumpla sus necesidades y requerimientos.

La finalidad de estos productos (menos cancelación) es de generar los Comprobantes Fiscales Digitales (**CFDi**) y el código bidimensional (**QRCode**), a través de Internet. Tiene los últimos requerimientos especificados para realiza todas las validaciones de los CFDi exigidas por el SAT.

A continuación hay una tabla comparativa de todos los productos que compone esta plataforma:

		Web Service	Web Service PDF	Conector Local	Web Conector	Web Service Portal
Características	<i>El Servicio forma el XML y lo sella</i>			✓	✓	
	<i>El cliente forma el XML y lo sella</i>	✓	✓			✓
	<i>El Servicio timbra el XML sellado</i>	✓	✓	✓	✓	✓
	<i>Guardar en Base de Datos Factorum</i>		✓	✓	✓	
	<i>El Servicio manda un Email de aviso al cliente de su factura</i>			✓	✓	
	<i>Manda a Impresión Automática la factura emitida</i>			✓		
	<i>El usuario tiene notificaciones por Email de folios por acabar</i>	✓	✓	✓	✓	✓
	<i>El Servicio hace la generación del PDF</i>		✓	✓	✓	✓
	<i>Uso de Bóveda Factorum</i>			✓	✓	
	<i>El Servicio tiene Conectividad con sistemas web</i>	✓	✓		✓	
	<i>El servicio tiene Conectividad con sistemas locales</i>	✓	✓	✓	✓	
	<i>El Servicio usa Esquema CFD</i>					
	<i>El Servicio usa Esquema CFDi</i>	✓	✓	✓	✓	✓
	<i>Hay un Portal de aplicación</i>					
<i>Hay un Portal de consulta</i>			✓	✓		
Entrada	<i>Usr, Pwd, RFC; Proporcionados para acceder</i>	✓	✓	✓	✓	✓
	<i>Archivo xml sellado construido por el usuario, es el contenido de lo que se quiere facturar</i>	✓	✓			✓

Archivo txt layout construido por el usuario, es el contenido de lo que se quiere facturar				✓	✓	
		Web Service	Web Service PDF	Conector Local	Web Conector	Web Service Portal
Salida o resultado	Archivo QRcode	✓	✓		✓	✓
	Archivo XML timbrado	✓	✓	✓	✓	✓
	Cadena Original Sello	✓	✓		✓	
	Fecha Timbrado	✓	✓		✓	
	No Certificado SAT	✓	✓		✓	
	Sello CDF	✓	✓		✓	
	Sello SAT	✓	✓		✓	
	UUID	✓	✓		✓	
	Tiempo de proceso		✓		✓	✓
	Archivo PDF		✓	✓	✓	✓
	Código de error		✓		✓	✓
	Mensaje de error		✓		✓	✓

La implementación del servicio se realiza a través del protocolo **https**. De esta forma, se utiliza un canal de comunicaciones seguro con el Proveedor de Servicios de Emisión de CFDI (PSECFDI).

Desde cualquier tecnología (cualquier lenguaje de programación que soporte llamadas a webservices mediante SOAP) pueda programar y desarrollar aplicaciones compatibles con el servicio e invocar los métodos, y tenga la facultad de recibir los valores devueltos por el servicio.

Existen desde el más bajo nivel, donde el usuario tiene la facultad de poder generar los datos de su factura en un xml y sellarlo, hasta los que pueden ser usados por los usuarios sin conocimientos en formar y sellar xml y el producto hace todo el trabajo

Los productos son:

- **Webservice**
- **Webservice PDF**
- **Webservice Portal**
- **Web Conector**
- **Conector Local**
- **Webservice Cancelación**

Cada producto tiene datos de entrada que necesita para procesarlos y al final entregar un resultado o datos de salida.

A continuación se resumen en una table comparativa los productos mencionados menos el de cancelación, ya que se maneja diferente a los demás.

WEB SERVICE

		Web Service
Características	El Servicio forma el XML y lo sella	
	El cliente forma el XML y lo sella	✓
	El Servicio timbra el XML sellado	✓
	Guardar en Base de Datos Factorum	
	El Servicio manda un Email de aviso al cliente de su factura	
	Manda a Impresión Automática la factura emitida	
	El usuario tiene notificaciones automáticas por correo electrónico de folios por acabar	✓
	El Servicio hace la generación del PDF	
	Uso de Bóveda Factorum	
	El Servicio tiene Conectividad con sistemas web	✓
	El servicio tiene Conectividad con sistemas locales	✓
	El Servicio usa Esquema CFD	
	El Servicio usa Esquema CFDi	✓
	Hay un Portal de aplicación	
Hay un Portal de consulta		
Entrada	Usr, Pwd, RFC; Proporcionados cuando se adquiere el producto	✓
	xml sellado construido por el usuario, es el contenido de lo que se quiere facturar	✓
	txt layout construido por el usuario, es el contenido de lo que se quiere facturar	
Salida	Archivo QRcode	✓
	Archivo XML timbrado	✓
	Cadena Original Sello	✓
	Fecha Timbrado	✓
	No Certificado SAT	✓
	Sello CDF	✓
	Sello SAT	✓
	UUID	✓
	Tiempo de proceso	
	Archivo PDF	
	Código de error	
Mensaje de error		

URL DE ACCESO AL SERVICIO

<https://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx>

FactorumCFDiService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [FactorumGenYaSellado](#)
- [FactorumGenYaSelladoConArchivo](#)
- [FactorumGenYaSelladoConArchivoTest](#)
- [FactorumGenYaSelladoTest](#)

Para acceder al web service definition lenguaje (WSDL:

<https://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx?WSDL>

MÉTODOS

En todos los métodos se necesita la información de un XML de entrada, donde estarán los datos que se quieren facturar, este XML debe de venir sellado

En cada llamada se puede generar un CFDi y el QRCode.

Todos los métodos tienen exactamente la misma estructura de salida, así como de validación, por eso las vamos a mencionar al acabar de explicar todos los métodos

Existen métodos para productivo y para test, se diferencian porque el usuario que usan los de test es de pruebas, esto sirve para una etapa de pruebas.

Existen tanto para productivo como para test, métodos que reciben lo que se desea facturar (el XML de entrada) en un string, o en un array de bytes.

Función	Descripción
FactorumGenYaSellado	Generación de timbre de CFDI. Permite obtener el timbre de la firma digital del CFD y el QRCode, a partir de los datos iniciales de tipo String
FactorumGenYaSelladoTest	Generación de CFDI en test. Funciona igual que el anterior, pero genera un comprobante con un timbre no válido, y no envía el comprobante al SAT. Esto es para pruebas
FactorumGenYaSelladoConArchivo	Generación de timbre de CFDI. Permite obtener el timbre de la firma digital del CFD y el QRCode, a partir de los datos iniciales de un archivo en formato XML
FactorumGenYaSelladoConArchivoTest	Generación de CFDI en test. Funciona igual que el anterior, pero genera un comprobante con un timbre no válido, y no envía el comprobante al SAT. Esto es para pruebas

FACTORUMGENYASELLADO

Este método es para producción, para datos reales.

DEFINICIÓN

```
public ReturnFactorumWS FactorumGenYaSellado (String usuario,  
String rfc,  
String password,  
String xml)
```

PARÁMETROS

- **usuario:** Usuario que va realizar la conexión, normalmente es el email
- **rfc:** RFC del usuario
- **password:** Contraseña del usuario que va realizar la conexión
- **xml:** Cadena de caracteres en formato string que tiene la información que se quiere facturar, datos como Emisor, Cliente, Detalle de los productos o servicios, total, etc. Formateado como XML, este XML debe de venir sellado

PETICIÓN Y RESPUESTA SOAP

<http://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx?op=FactorumGenYaSellado>

FACTORUMGENYASELLADOTEST

Este método es para pruebas, debe de usar el usuario de pruebas.

DEFINICIÓN

```
public ReturnFactorumWS FactorumGenYaSelladoTest (String usuario,  
String rfc,  
String password,  
String xml)
```

PARÁMETROS

- **usuario, rfc:** y **password:** de pruebas
- **xml:** Cadena de caracteres en formato XML que tiene la información que se quiere facturar, datos como Emisor, Cliente, Detalle de los productos o servicios, total, etc. Formateado como XML, este XML debe de venir sellado

PETICIÓN Y RESPUESTA SOAP

<http://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx?op=FactorumGenYaSelladoTest>

EJEMPLOS PARA FACTORUMGENYASELLADO Y FACTORUMGENYASELLADOTEST

Código del cliente:

Para usar el método deseado cambie solamente el nombre del método, aquí le presentamos FactorumGenYaSelladoTest

```

'Entrada de la información de la venta en formato XML
Dim XmlVta As String = XmlGenerado
'Resultado del CFDi en formato XML
Dim signedXml As String
'Resultado del QRcode
Dim Qrcode As Byte()
'Resultado del CFDi en formato XML
Dim FilesignedXml As Byte()

Dim ws As New FactorumCFDiService
Dim resWS As New ReturnFactorumWS

Try
'llamar al WS
    resWS = ws.FactorumGenYaSelladoTest (userWS, rfcWS, pwdWS,
XmlVta)
'Regresa el CFDi sellado y timbrado
    signedXml = resWS.ReturnStringXML
'Regresa el CFDi en archivo
    FilesignedXml = resWS.ReturnFileXML
Guardar físicamente el xml en un archivo
    Dim Stream1 As Stream = New FileStream("C:\a.xml", _
                                           FileMode.Create)
    Stream1.Write(FilesignedXml, 0, FilesignedXml.Length)
    Stream1.Flush()
    Stream1.Close()
'Regresa el QRcode
    Qrcode = resWS.ReturnFileQRCode
Guardar físicamente el QRcode en un gif
    Dim stream2 As Stream = New FileStream("C:\a.gif", _
                                           FileMode.Create)
    stream2.Write(Qrcode, 0, Qrcode.Length)
    stream2.Flush()
    stream2.Close()

Catch excep As Exception

    Me.Mensaje.Text = excep.Message

End Try

```

FACTORUMGENYASELLADOCONARCHIVO

Este método es para producción, para datos reales. Es parecido a el método de FactorumGenYaSellado, solo que el último parámetro es un array de bytes (un archivo)

DEFINICIÓN

```
public ReturnFactorumWS FactorumGenYaSelladoConArchivo (String usuario,
String rfc,
String password,
Byte[] xml)
```

PARÁMETROS

- **usuario:** Usuario que va realizar la conexión, normalmente es el email
- **rfc:** RFC del usuario
- **password:** Contraseña del usuario que va realizar la conexión
- **xml:** array de bytes que contiene la lectura de un archivo en formato XML que tiene la información que se quiere facturar, datos como Emisor, Cliente, Detalle de los productos o servicios, total, etc. Formateado como XML, este XML debe de venir sellado

PETICIÓN Y RESPUESTA SOAP

<http://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx?op=FactorumGenYaSelladoConArchivo>

FACTORUMGENYASELLADOCONARCHIVOEST

Este método es para pruebas, debe de usar el usuario de pruebas.

DEFINICIÓN

```
public ReturnFactorumWS FactorumGenYaSelladoConArchivoTest (String usuario,
String rfc,
String password,
Byte[] xml)
```

PARÁMETROS

- **usuario, rfc:** y **password:** de Pruebas
- **xml:** array de bytes que contiene la lectura de un archivo en formato XML que tiene la información que se quiere facturar, datos como Emisor, Cliente, Detalle de los productos o servicios, total, etc. Formateado como XML, este XML debe de venir sellado

PETICIÓN Y RESPUESTA SOAP

<http://www.factorumweb.com/FactorumWSv32/FactorumCFDiService.asmx?op=FactorumGenYaSelladoConArchivoTest>

EJEMPLOS

Código del cliente:

Para usar el método deseado cambie solamente el nombre del método, aquí le presentamos FactorumYaSelladoConArchivoTest

```
Lectura de la información de la venta en formato XML
Dim FileXmlClient As Byte()
```

```

Dim stream As Stream = New FileStream("C:Prueba.xml", FileMode.Open,
FileAccess.Read, FileShare.Read)
    Dim streamReader As StreamReader = New StreamReader(stream,
Encoding.Default)

Dim datalen As Integer = stream.Length
ReDim FileXmlClient (datalen)

stream.Seek(0, SeekOrigin.Begin)
stream.Read(FileXmlClient, 0, datalen)
stream.Close()

'Resultado del CFDi en formato XML
Dim FilesignedXml As Byte()
'Resultado del QRcode
Dim Qrcode As Byte()

'llamar al WS
*****

Dim ws As New FactorumCFDiService
Dim resWS As New ReturnFactorumWS

Try
'llamar al WS
    resWS = ws.FactorumYaSelladoConArchivoTest (userWS, _
                                                rfcWS, _
                                                pwdWS, _
                                                FileXmlClient)

'Regresa el CFDi sellado y timbrado
    FilesignedXml = resWS.ReturnFileXML
Guardar físicamente el xml en un archivo
    Dim Stream1 As Stream = New FileStream("C:\a.xml", _
                                                FileMode.Create)
    Stream1.Write(FilesignedXml, 0, FilesignedXml.Length)
    Stream1.Flush()
    Stream1.Close()

'Regresa el QRcode
    Qrcode = resWS.ReturnFileQRCode
Guardar físicamente el QRcode en un gif
    Dim stream2 As Stream = New FileStream("C:\a.gif", _
                                                FileMode.Create)
    stream2.Write(Qrcode, 0, Qrcode.Length)
    stream2.Flush()
    stream2.Close()
Catch excep As Exception
    Me.Mensaje.Text = excep.Message
End Try

```

RESPUESTA DE LOS MÉTODOS

Todos los métodos anteriores regresan un tipo de datos: **ReturnFactorumWS** el cual es definido como sigue:

```
public class ReturnFactorumWS
{
    public String ReturnStringXML;
    public Byte[] ReturnFileXML;
    public Byte[] ReturnFileQRCode;
    public dateTime FechaTimbrado
    public String UUID
    public String SelloCFD
    public String NoCertificadoSAT
    public String SelloSAT
    public String CadenaOriginalSello
}
```

- **ReturnStringXML:** Es un string que contiene el CFDi, es en formato XML.
- **ReturnFileXML:** Es el archivo que contiene el CFDi, es en formato XML.
- **ReturnFileQRCode:** Es el archivo del Código Bidimensional, cuando se recibe, se puede guardar en formato GIF.
- **FechaTimbrado:** Regresa la fecha en la que fue timbrado el CFDi
- **UUID:** Regresa el timbre del SAT. El código se forma por una cadena compuesta por 32 dígitos hexadecimales
- **SelloCFD:** regresa el sello del CFD proporcionado por el SAT.
- **NoCertificadoSat:** Regresa el Número de Certificado proporcionado por el SAT.
- **SelloSat:** Regresa el Sello que proporciona el SAT.
- **CadenaOriginalSello:** Regresa la cadena original con la que fue sellado el documento antes de enviarlo a Timbrar

VALIDACIONES QUE CONTIENE EL SERVICIO

Para todos los métodos:

- Valida que el nombre de Usuario y Password coincidan y estén en la base de datos.
- Valida que la estructura del parámetro de entrada (**String xml**) cumple con las especificaciones técnicas.
- Valida que el usuario no sea de tipo DEMO, en caso de los que no tengan terminación TEST
- Valida que los folios contratados estén vigentes en caso de productivo
- Valida los diferentes estatus que el usuario pueda tener (cancelado, baja, activo)
- Envía un email indicando cuántos folios quedan por usar, cuando se llegue a una línea de 15% restantes, 10% restantes y 5% restantes

Para poder detectar si hubo un error, es necesario utilizar un manejo de errores dentro de la programación, y poder tomar decisiones dentro del código

LISTA DE CÓDIGOS DE ERROR

El SAT ha publicado una lista estándar de códigos de error (códigos del 301 al 403) aplicables a las respuestas no satisfactorias.

El servicio detecta estos errores, y los regresa. También verifica otras situaciones de error directamente relacionadas con la aplicación de Factorum.

ERRORES
ESTATUS DE USUARIO INCORRECTO
ESTATUS IRRECONOCIBLE
PASSWORD NO COINCIDE
USUARIO Y PASSWORD INCORRECTO, NO ENCONTRADO EN BD
PASSWORD NO PROPORCIONADO
USUARIO NO PROPORCIONADO
ERROR AL GUARDAR EN DISCO EL ARCHIVO GIF
ERROR AL GUARDAR EN DISCO EL ARCHIVO XML

WEBSERVICE PDF

		Webservice PDF
Características	El Servicio forma el XML y lo sella	
	El cliente forma el XML y lo sella	✓
	El Servicio timbra el XML sellado	✓
	Guardar en Base de Datos Factorum	✓
	El Servicio manda un Email de aviso al cliente de su factura	
	Manda a Impresión Automática la factura emitida	
	El usuario tiene notificaciones automáticas por correo electrónico de folios por acabar	✓
	El Servicio hace la generación del PDF	✓
	Uso de Bóveda Factorum	
	El Servicio tiene Conectividad con sistemas web	✓
	El servicio tiene Conectividad con sistemas locales	✓
	El Servicio usa Esquema CFD	
	El Servicio usa Esquema CFDi	✓
	Hay un Portal de aplicación	
	Hay un Portal de consulta	
Entrada	Usr, Pwd, RFC; Proporcionados cuando se adquiere el producto	✓
	xml sellado construido por el usuario, es el contenido de lo que se quiere facturar	✓
	txt layout construido por el usuario, es el contenido de lo que se quiere facturar	
Salida	Archivo QRcode	✓
	Archivo XML timbrado	✓
	Cadena Original Sello	✓
	Fecha Timbrado	✓
	No Certificado SAT	✓
	Sello CDF	✓
	Sello SAT	✓
	UUID	✓
	Tiempo de proceso	✓
	Archivo PDF	✓
	Código de error	✓
	Mensaje de error	✓

URL DE ACCESO AL SERVICIO

<https://www.factorumweb.com/FactorumWSv32pdf/Factorumwspdf.aspx>

FactorumWSpdf

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [wsFactorumWSpdf](#)

Para acceder al web service definition lenguaje (WSDL):

<https://www.factorumweb.com/FactorumWSpdfv32/Factorumwspdf.aspx?WSDL>

MÉTODO

Función	Descripción
wsFactorumWSpdf	Generación de timbre de CFDI. Permite obtener el timbre de la firma digital del CFD, el archivo PDF y el QRCode, a partir de los datos iniciales de un archivo en formato XML

En cada llamada se puede generar un CFDi y el QRCode.

DEFINICIÓN

```
public ReturnFactorumService wsFactorumWSpdf (String usuario,
String rfc,
String password,
String NombreArchivo,
Byte[] xml,
Boolean FlagTest)
```

PARÁMETROS

- **usuario:** Usuario que va realizar la conexión, normalmente es el email
- **rfc:** RFC del usuario
- **password:** Contraseña del usuario que va realizar la conexión
- **NombreArchivo:** Se le indica cuál es el nombre para el PDF resultante, el nombre físico del archivo. Sin extensión y sin ruta.
- **xml:** array de bytes, que es la lectura del Archivo en formato XML que tiene la información que se quiere facturar, datos como Emisor, Cliente, Detalle de los productos o servicios, total, etc. Formateado como XML, este XML debe de venir sellado
- **FlagTest:** True= ambiente de pruebas, False = Ambiente productivo. Cuando tenga el valor de true, deberá de poner los datos del usuario de pruebas.

PETICIÓN Y RESPUESTA SOAP

<http://www.factorumweb.com/FactorumWSvpdfv32/Factorumwspdf.asmx?op=wsFactorumWSpdf>

RESPUESTA DEL MÉTODO

Regresa un tipo de datos: `ReturnFactorumService` el cual es definido como sigue:

```
Public Class ReturnFactorumService
    Public StringXML As String = ""
    Public ByteXML As Byte()
    Public ByteQRCode As Byte()
    Public BytePDF As Byte()
    Public FechaTimbrado As System.DateTime
    Public UUID As String = ""
    Public SelloCFD As String = ""
    Public NoCertificadoSAT As String = ""
    Public SelloSAT As String = ""
    Public CadenaOriginalSello As String = ""
    Public TiempoProceso As String = ""
End Class
```

- **StringXML:** Es un string que contiene el CFDi, es en formato XML.
- **ByteXML:** Es el archivo que contiene el CFDi, es en formato XML.
- **ByteQRCode:** Es el archivo del Código Bidimensional, cuando se recibe, se puede guardar en formato GIF.
- **BytePDF:** Es el archivo de la factura lista para imprimir en formato PDF
- **FechaTimbrado:** Regresa la fecha en la que fue timbrado el CFDi
- **UUID:** Regresa el timbre del SAT. El código se forma por una cadena compuesta por 32 dígitos hexadecimales
- **SelloCFD:** regresa el sello del CFD proporcionado por el SAT.
- **NoCertificadoSat:** Regresa el Número de Certificado proporcionado por el SAT.
- **SelloSat:** Regresa el Sello que proporciona el SAT.
- **CadenaOriginalSello:** Regresa la cadena original con la que fue sellado el documento antes de enviarlo a Timbrar
- **TiempoProceso:** Regresa en segundos, cuánto tiempo tardo la transacción

VALIDACIONES QUE CONTIENE EL SERVICIO

- Valida que el nombre de Usuario y Password coincidan y estén en la base de datos.
- Valida que la estructura del parámetro de entrada (`Byte [] xml`) cumple con las especificaciones técnicas.
- Valida que el usuario no sea de tipo DEMO
- Valida que los folios contratados estén vigentes
- Valida los diferentes estatus que el usuario pueda tener

EJEMPLOS

Código del cliente:

```
Dim FileXmlClient As Byte()
```

```

Dim stream As Stream = New FileStream("C:Prueba.xml", FileMode.Open,
FileAccess.Read, FileShare.Read)
    Dim streamReader As StreamReader = New StreamReader(stream,
Encoding.Default)

Dim datalen As Integer = stream.Length
ReDim FileXmlClient (datalen)

stream.Seek(0, SeekOrigin.Begin)
stream.Read(FileXmlClient, 0, datalen)
stream.Close()

Dim NombreArchivoFactura As String="Fac0001"

Try

    wsResultado = wsConector.wsFactorumWSpdf(Usuario, RFC, PWD, _
        NombreArchivoFactura, _
        FileXmlClient, _
        FlagTest)
    'el resultado esta en la clase de wsResultado
    'Guardar en archivo los arrays de bytes
    func.EscribeBytesEnArchivo(Path & NombreArchivoFactura &
".PDF", wsResultado.BytePDF)
    func.EscribeBytesEnArchivo(Path & NombreArchivoFactura &
".XML", wsResultado.ByteXML)
    func.EscribeBytesEnArchivo(Path & NombreArchivoFactura &
".GIF", wsResultado.ByteQRCode)
    'Mostrar en pantalla los datos
    func.EscribeEnPantalla("Los archivos GIF, XML, y PDF estan
en: : ", Path)
    func.EscribeEnPantalla("CadenaOriginalSello: ",
wsResultado.CadenaOriginalSello)
    func.EscribeEnPantalla("FechaTimbrado: ",
wsResultado.FechaTimbrado)
    func.EscribeEnPantalla("NoCertificadoSAT: ",
wsResultado.NoCertificadoSAT)
    func.EscribeEnPantalla("SelloCFD: ", wsResultado.SelloCFD)
    func.EscribeEnPantalla("SelloSAT: ", wsResultado.SelloSAT)
    func.EscribeEnPantalla("UUID: ", wsResultado.UUID)
    func.EscribeEnPantalla("Todo el proceso tardó: ",
wsResultado.TiempoProceso & " segundos")
Catch wsError As Exception
    If wsError.Message.Contains("|") Then
        func.EscribeEnPantalla("-- ERROR CODIGO-- ",
wsError.Message.Split("|")(1))
        func.EscribeEnPantalla("-- ERROR DESCRIPCION-- ",
wsError.Message.Split("|")(2))
    Else
        func.EscribeEnPantalla("-- ERROR -- ", wsError.Message)
    End If
End Try

```

LISTA DE CÓDIGOS DE ERROR

El SAT ha publicado una lista estándar de códigos de error (códigos del 301 al 403) aplicables a las respuestas no satisfactorias.

El servicio detecta estos errores, y los regresa. También verifica otras situaciones de error directamente relacionadas con la aplicación de Factorum.

ERRORES
ESTATUS DE USUARIO INCORRECTO
ESTATUS IRRECONOCIBLE
PASSWORD NO COINCIDE
USUARIO Y PASSWORD INCORRECTO, NO ENCONTRADO EN BD
PASSWORD NO PROPORCIONADO
USUARIO NO PROPORCIONADO
ERROR AL GUARDAR EN DISCO EL ARCHIVO GIF
ERROR AL GUARDAR EN DISCO EL ARCHIVO XML
FACTURA YA HECHA (FACTURA TIMBRADA ANTERIORMENTE)

Para poder detectar si hubo un error, es necesario utilizar un manejo de errores dentro de la programación, y poder tomar decisiones dentro del código

CONECTOR LOCAL

		Conector Local
Características	El Servicio forma el XML y lo sella	✓
	El Servicio timbra el XML sellado	✓
	Guardar en Base de Datos Factorum	✓
	El Servicio manda un Email de aviso al cliente de su factura	✓
	Manda a Impresión Automática la factura emitida	✓
	El usuario tiene notificaciones automáticas por correo electrónico de folios por acabar	✓
	El Servicio hace la generación del PDF	✓
	Uso de Bóveda Factorum	✓
	El Servicio tiene Conectividad con sistemas web	
	El servicio tiene Conectividad con sistemas locales	✓
	El Servicio usa Esquema CFD	
	El Servicio usa Esquema CFDi	✓
	Hay un Portal de aplicación	
	Hay un Portal de consulta	✓
Entrada	Usr, Pwd, RFC; Proporcionados cuando se adquiere el producto xml sellado construido por el usuario, es el contenido de lo que se quiere facturar	✓
	txt layout construido por el usuario, es el contenido de lo que se quiere facturar	✓
Salida	Archivo QRcode	
	Archivo XML timbrado	✓
	Cadena Original Sello	
	Fecha Timbrado	
	No Certificado SAT	
	Sello CDF	
	Sello SAT	
	UUID	
	Tiempo de proceso	
	Archivo PDF	✓
	Código de error	
Mensaje de error		

Es un programa que se instala en la máquina del cliente, y espera recibir archivos de texto que contienen la información a facturar, bajo un esquema (layout) acordado, de tal manera que cada archivo depositado es procesado usando internamente nuestro webservice, y se regresa la respuesta en un directorio donde estarán los PDF y XML ya timbrados.

Como requisito, debe de tener instalado el ADOBE READER, para poder imprimir los PDF.

Un apartado importante son el archivo de configuración, y las carpetas que se manejan

EXPLICACIÓN DEL ARCHIVO DE CONFIGURACIÓN:

El archivo de configuración es un archivo en formato XML, donde le podemos modificar parámetros en su nodo de <appSettings>

<add key="**SEGUNDOS**" value="5" /> es el tiempo que se checará en el directorio de entrada, si hay algún archivo a procesar.

<add key="**DEBUG**" value="N" /> Si se le pone "S" permite sacar una ventana para "testear" las funciones. N = actúa de manera silenciosa procesando los archivos.

<add key="**METODOTEST**" value="S" /> Para que el timbrado sea mediante un ambiente de pruebas. N = Productivo

<add key="**ADOBE**" value="C:\Program Files (x86)\Adobe\Reader 9.0\Reader\AcroRd32.exe" /> Ruta EXACTA de donde esté instalado el Adobe Reader, esto es para la impresión automática.

<add key="**RFC**" value="AAA010101AAA" /> Datos del usuario de prueba, una vez entregado el CER y el KEY y el pwd del SAT, se pondrán sus datos

<add key="**USR**" value="prueba1@factorum.com.mx" /> Datos del usuario de prueba, una vez entregado el CER y el KEY y el pwd del SAT, se pondrán sus datos

<add key="**PWD**" value="prueba2011" /> Datos del usuario de prueba, una vez entregado el CER y el KEY y el pwd del SAT, se pondrán sus datos

Estos parámetros cambiarlos de acuerdo a como haya instalado sus carpetas, o sea, debe de poner la ruta exacta en cada una de ellas:

<add key="**PATH_PDF**" value="C:\Archivos_PDF" /> Ruta de directorios (cambiar a donde se vayan a realizar las pruebas)

<add key="**PATH_IN**" value="C:\Archivos_Entrada" /> Ruta de directorios (cambiar a donde se vayan a realizar las pruebas)

<add key="**PATH_ERROR**" value="C:\Archivos_Error" /> Ruta de directorios (cambiar a donde se vayan a realizar las pruebas)

<add key="**PATH_OUT**" value="C:\Archivos_Salida" /> Ruta de directorios (cambiar a donde se vayan a realizar las pruebas)

EXPLICACION DE LAS CARPETAS

ARCHIVOS DE ENTRADA:

Los Archivos DAT que están ahí no borrarlos.

Mapping: Sirve para hacer match entre el Layout y nuestra DB

Número.dat: son las direcciones respectivamente (siguiendo el layout original)

0,1: Direccion por sucursales que esta siendo indicado en el Layout

ARCHIVOS PDF

La estructura que manejamos para los subdirectorios del **PDF** es la siguiente:

Dado el siguiente ejemplo de una carpeta:

```
\Archivos_PDF\0\05\13
```

Elementos:

\Archivos_PDF: Definido en el archivo Config

\0 : Número de sucursal, este sale del layout en la coordenada E01:10

\05: Es el mes, que sale de la extracción de la fecha en el Layout es la coordenada E01:2. NOTA: el formato debe de ser "yyyy-mm-dd"

\13: Es el tipo del documento, en el Layout está en la coordenada E01:9

Entonces queda la agrupación por Sucursal, y por cada sucursal se agrupa por mes, y dentro de cada mes, por tipo de documento.

ARCHIVOS SALIDA

La estructura que manejamos es la siguiente:

Dado el siguiente ejemplo de una carpeta:

```
\Archivos_Salida\0
```

Elementos:

\Archivos_Salida: Definido en el archivo Config

\0 : Número de sucursal, este sale del layout en la coordenada E01:10 Aquí se deja una copia del archivo que se proceso, es decir, se quita del directorio de Entrada y se pasa al de Salida.

ARCHIVOS ERROR

Aquí se dejan los TXT que no se procesaron porque sucedió algún error. Para saber qué error ocurrió existe un archivo que se llama LogError.txt, el cual va guardando entradas por cada error que ocurre.

Un ejemplo de entrada de error es:

```
Folio Factorum: 17330|Serie/Folio: U1/666|Paso 1: 1|Paso 2: 2|Paso 3: 0|Paso 4: 0|Paso 5: 0|Paso 6: 0|Nombre Arch: U1AUTOS|Mensaje Error: Error FormaXmlValido:
```

```
Server was unable to process request. ---> Object reference not set to an instance of an object.
```

Separados por pipes contienen los siguientes datos:

- Indica el ID de la factura, si es que alcanzó a grabar en la base de datos de factorum
- Indica la Serie y follio de la factura a procesar
- El estatus de todos los pasos, hay seis pasos, los estatus pueden ser:
 - 1 es Ok
 - 2 es Error

- 0 = no entro al paso
- Nombre del Archivo TXT que se procesa
- Mensaje del error

EXPLICACION DEL LAYOUT:

Al inicio de un registro hay un identificador de registro (E01, EA1, D01..etc.) en cada registro hay campos numerados.

Cada campo debe de ser separado por Pipes

Cuando no haya valor en un campo, se debe de poner vacio, por ejemplo EA1 | | |

Hay Registros que se pueden repetir dentro del archivo, por ejemplo el detalle de la factura (D01), para estos campos, se indican en el layout mediante un corchete.

Nota, es importante que el primer campo tenga el Folio y Serie, separados por un undersocre, por ejemplo:

E01 | 0001_A... Esto significa que es folio 0001 y serie A.

EXPLICACION DE LOS ARCHIVOS 0.DAT, 1.DAT, ETC..

Estos archivos representan la identificacion de sucursales, ahí se genera cada archivo (menos el 0) para identificar que sucursal fue la que hizo la factura. Es en realidad los datos del cliente emisor.

La manera de hacerlos es la siguiente:

Separado por pipes, hay que poner en este orden los siguientes campos:

- [RFC]
- [Razonsocial]
- [Direccion1]
- [Colonia]
- [NumInterior]
- [NumExterior]
- [CP]
- [Municipio]
- [Localidad]
- [Estado]
- [Pais]

Si no hubiera sucursales, entonces se manda el 0 en el layout, en el E01 en el campo 10, si hubiera una sucursal, se manda el numero del archivo que le toco.

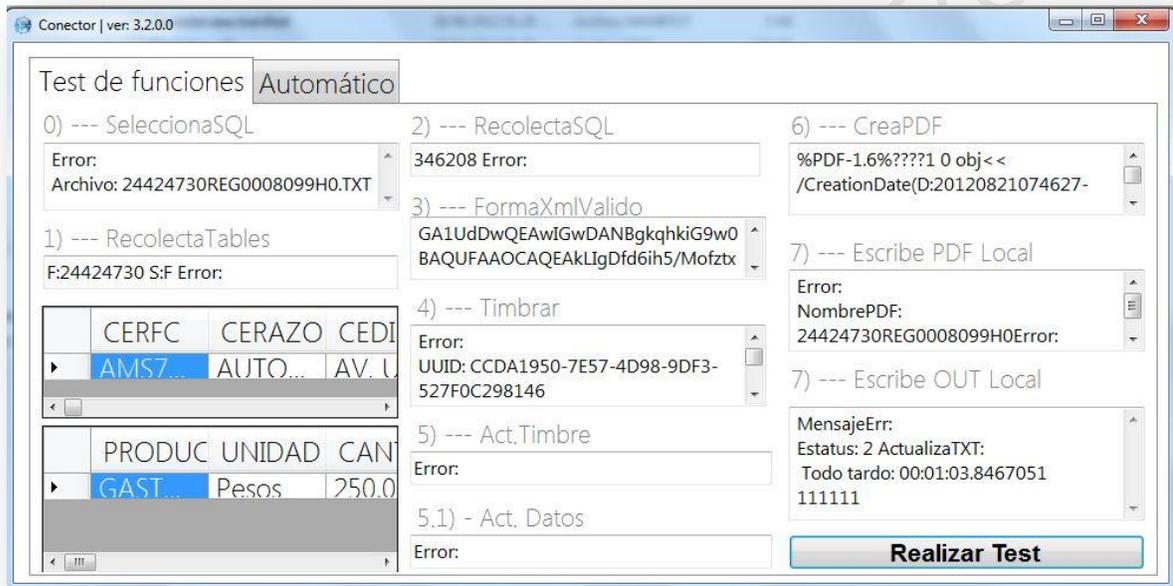
EXPLICACION DEL FLUJO DE INFORMACIÓN.

Esto se procesa de la siguiente manera. Usted deja el archivo TXT en la carpeta de ENTRADA (eso se pone en el config), el conector esta monitoreando esta carpeta en determinados segundos (eso se pone en el config) y en cuanto haya archivos ahí, los procesa inmediatamente, haciendo el sellado, el timbre, y pocos segundos después, deja el resultado en la carpeta de SALIDA y de PDF (eso se pone en el config), si es que todo salió bien, sino, los deja en la carpeta de ERROR (eso se pone en el config) con su entrada correspondiente en el log de error.

- En el paso 0 se busca el archivo TXT candidato para procesar

- En el paso 1 se lee y se entiende ese archivo, como resultado se presenta el Folio y la serie y el encabezado y detalle que tiene el txt
- En el paso 2 se graba todo en la base de datos, y regresa un folio de Factorum
- En el paso 3 se genera un XML sellado y se presenta el array de bytes de ese XML
- En el paso 4 se manda esa informacion al SAT y presentamos el UUID
- En el paso 5 se actualiza en nuestra base de datos los datos regresados del SAT
- En el paso 5.1 se graban datos adicionales en nuestra base de datos
- En el paso 6 recopila toda la informacion y crea el PDF, presentamos el array de bytes que representa la informacion
- En el paso 7 se escribe el PDF en forma local, presentamos el nombre del archivo
- En el paso 7(bis) se escriben los logs locales

Esta es la pantalla mostrando todos los pasos en modo de DEBUG = S



WEB CONECTOR

		Web Conector
Características	El Servicio forma el XML y lo sella	✓
	El Servicio timbra el XML sellado	✓
	Guardar en Base de Datos Factorum	✓
	El Servicio manda un Email de aviso al cliente de su factura	✓
	Manda a Impresión Automática la factura emitida	
	El usuario tiene notificaciones automáticas por correo electrónico de folios por acabar	✓
	El Servicio hace la generación del PDF	✓
	Uso de Bóveda Factorum	✓
	El Servicio tiene Conectividad con sistemas web	✓
	El servicio tiene Conectividad con sistemas locales	✓
	El Servicio usa Esquema CFD	
	El Servicio usa Esquema CFDi	✓
	Hay un Portal de aplicación	
	Hay un Portal de consulta	✓
Entrada	Usr, Pwd, RFC; Proporcionados cuando se adquiere el producto xml sellado construido por el usuario, es el contenido de lo que se quiere facturar	✓
	txt layout construido por el usuario, es el contenido de lo que se quiere facturar	✓
Salida	Archivo QRcode	✓
	Archivo XML timbrado	✓
	Cadena Original Sello	✓
	Fecha Timbrado	✓
	No Certificado SAT	✓
	Sello CDF	✓
	Sello SAT	✓
	UUID	✓
	Tiempo de proceso	✓
	Archivo PDF	✓
	Código de error	✓
Mensaje de error	✓	

URL DE ACCESO AL SERVICIO

<https://www.factorumweb.com/conectorwebv32/Factorumwebconector.asmx>

FactorumWebConector

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [wsFactorumWebConector](#)

Para acceder al web service definition lenguaje (WSDL):

<https://www.factorumweb.com/conectorwebv32/Factorumwebconector.asmx?WSDL>

MÉTODOS

Función	Descripción
wsFactorumWebConector	Generación de timbre de CFDI. Permite obtener el timbre de la firma digital del CFD, el archivo PDF y el QRCode, a partir de los datos iniciales de un archivo en formato txt apoyado en un Layout

En cada llamada se puede generar un CFDi y el QRCode.

DEFINICIÓN

```
public ReturnFactorumService wsFactorumWSpdf (String usuario,
String rfc,
String password,
String NombreArch,
Byte[] txtLayout,
Boolean FlagTest)
```

PARÁMETROS

- **usuario:** Usuario que va realizar la conexión, normalmente es el email
- **rfc:** RFC del usuario
- **password:** Contraseña del usuario que va realizar la conexión
- **NombreArch:** Es el nombre físico del archivo que tendrá la base de datos factorum para esta factura.
- **txtLayout:** Array de bytes, que es la lectura de un archivo en formato texto que se forma mediante un layout, y representa los datos de una factura.
- **FlagTest:** True= ambiente de pruebas, False = Ambiente productivo

EXPLICACION DEL LAYOUT:

Para formar el archivo TXT que representa a la factura, se debe de seguir un layout.

Al inicio de un registro hay un identificador de registro (E01, EA1, D01..etc.) en cada registro hay campos numerados.

Cada campo debe de ser separado por Pipes

Cuando no haya valor en un campo, se debe de poner vacío, por ejemplo EA1 | | |

Hay Registros que se pueden repetir dentro del archivo, por ejemplo el detalle de la factura (D01), para estos campos, se indican en el layout mediante un corchete.

Nota, es importante que el primer campo tenga el Folio y Serie, separados por un underscore, por ejemplo:

E01|0001_A... Esto significa que es folio 0001 y serie A.

PETICIÓN Y RESPUESTA SOAP

Para conocer el request y response se puede ver mediante esta liga:

<http://www.factorumweb.com/conectorwebv32/Factorumwebconector.asmx?op=wsFactorumWebConector>

RESPUESTA

Regresa un tipo de datos: `ReturnFactorumService` el cual es definido como sigue:

```
Public Class ReturnFactorumService
    Public StringXML As String = ""
    Public ByteXML As Byte()
    Public ByteQRCode As Byte()
    Public BytePDF As Byte()
    Public FechaTimbrado As System.DateTime
    Public UUID As String = ""
    Public SelloCFD As String = ""
    Public NoCertificadoSAT As String = ""
    Public SelloSAT As String = ""
    Public CadenaOriginalSello As String = ""
    Public TiempoProceso As String = ""
    Public CodigError As Integer = 0
    Public DescripcionError As String = ""

    End Class
```

- **StringXML:** Es un string que contiene el CFDi, es en formato XML.
- **ByteXML:** Es el archivo que contiene el CFDi, es en formato XML.
- **ByteQRCode:** Es el archivo del Código Bidimensional, cuando se recibe, se puede guardar en formato GIF.
- **BytePDF:** Es el archivo de la factura lista para imprimir en formato PDF
- **FechaTimbrado:** Regresa la fecha en la que fue timbrado el CFDi
- **UUID:** Regresa el timbre del SAT. El código se forma por una cadena compuesta por 32 dígitos hexadecimales
- **SelloCFD:** regresa el sello del CFD proporcionado por el SAT.
- **NoCertificadoSat:** Regresa el Número de Certificado proporcionado por el SAT.
- **SelloSat:** Regresa el Sello que proporciona el SAT.
- **CadenaOriginalSello:** Regresa la cadena original con la que fue sellado el documento antes de enviarlo a Timbrar

- **TiempoProceso:** Regresa en segundos, cuánto tiempo tardo la transacción
- **CodigoError:** Regresa el código de error. 0=Sin error
- **DescripcionError:** Regresa la descripción del error, si es que hubiera.

VALIDACIONES QUE CONTIENE EL SERVICIO

- Valida que el nombre de Usuario y Password coincidan y estén en la base de datos.
- Valida que la estructura del parámetro de entrada cumple con las especificaciones del layout.
- Valida que los folios contratados estén vigentes
- Valida los diferentes estatus que el usuario pueda tener

EJEMPLO

Código del cliente:

```
'Lee el archivo generado del layout
Dim txtLayout As Byte()

Dim stream As Stream = New FileStream("C:Prueba.txt", FileMode.Open,
FileAccess.Read, FileShare.Read)
    Dim streamReader As StreamReader = New StreamReader(stream,
Encoding.Default)

Dim datalen As Integer = stream.Length
ReDim txtLayout (datalen)

stream.Seek(0, SeekOrigin.Begin)
stream.Read(txtLayout, 0, datalen)
stream.Close()
    Try

        wsResultado = wsConector.wsFactorumWebConector(Usr, RFC, PWD, _
NombreArchivoFactura, txtLayout, _
FlagTest)
        'el resultado esta en la clase de wsResultado
        'Guardar en archivo los arrays de bytes
        func.EscribeBytesEnArchivo(Path & NombreArchivoFactura & ".PDF",
wsResultado.BytePDF)
        func.EscribeBytesEnArchivo(Path & NombreArchivoFactura & ".XML",
wsResultado.ByteXML)
        func.EscribeBytesEnArchivo(Path & NombreArchivoFactura & ".GIF",
wsResultado.ByteQRCode)
        'Mostrar en pantalla los datos
        func.EscribeEnPantalla("Los archivos GIF, XML, y PDF estan en: :
", Path)
        func.EscribeEnPantalla("CadenaOriginalSello: ",
wsResultado.CadenaOriginalSello)
        func.EscribeEnPantalla("FechaTimbrado: ",
wsResultado.FechaTimbrado)
        func.EscribeEnPantalla("NoCertificadoSAT: ",
wsResultado.NoCertificadoSAT)
        func.EscribeEnPantalla("SelloCFD: ", wsResultado.SelloCFD)
        func.EscribeEnPantalla("SelloSAT: ", wsResultado.SelloSAT)
        func.EscribeEnPantalla("UUID: ", wsResultado.UUID)
        func.EscribeEnPantalla("Todo el proceso tardo: ",
wsResultado.TiempoProceso & " segundos")
```

```

Catch wsError As Exception
    If wsError.Message.Contains("|") Then
        func.EscribeEnPantalla("-- ERROR CODIGO-- ",
wsError.Message.Split("|")(1))
        func.EscribeEnPantalla("-- ERROR DESCRIPCION-- ",
wsError.Message.Split("|")(2))
    Else
        func.EscribeEnPantalla("-- ERROR -- ", wsError.Message)
    End If
End Try

```

LISTA DE CÓDIGOS DE ERROR

El SAT ha publicado una lista estándar de códigos de error (códigos del 301 al 403) aplicables a las respuestas no satisfactorias.

El servicio detecta estos errores, y los regresa. También verifica otras situaciones de error directamente relacionadas con la aplicación de Factorum.

ERRORES
ESTATUS DE USUARIO INCORRECTO
ESTATUS IRRECONOCIBLE
PASSWORD NO COINCIDE
USUARIO Y PASSWORD INCORRECTO, NO ENCONTRADO EN BD
PASSWORD NO PROPORCIONADO
USUARIO NO PROPORCIONADO
ERROR AL GUARDAR EN DISCO EL ARCHIVO GIF
ERROR AL GUARDAR EN DISCO EL ARCHIVO XML
FACTURA YA HECHA (FACTURA TIMBRADA ANTERIORMENTE)

Para poder detectar si hubo un error, es necesario utilizar un manejo de errores dentro de la programación, y poder tomar decisiones dentro del código

WEBSERVICE PORTAL

		WebService Portal
Características	El Servicio forma el XML y lo sella	
	El Servicio timbra el XML sellado	✓
	Guardar en Base de Datos Factorum	
	El Servicio manda un Email de aviso al cliente de su factura	
	Manda a Impresión Automática la factura emitida	
	El usuario tiene notificaciones automáticas por correo electrónico de folios por acabar	✓
	El Servicio hace la generación del PDF	✓
	Uso de Bóveda Factorum	
	El Servicio tiene Conectividad con sistemas web	
	El servicio tiene Conectividad con sistemas locales	
	El Servicio usa Esquema CFD	
	El Servicio usa Esquema CFDi	✓
	Hay un Portal de aplicación	
	Hay un Portal de consulta	
Entrada	Usr, Pwd, RFC; Proporcionados cuando se adquiere el producto	✓
	xml sellado construido por el usuario, es el contenido de lo que se quiere facturar	✓
	txt layout construido por el usuario, es el contenido de lo que se quiere facturar	
Salida	Archivo QRcode	✓
	Archivo XML timbrado	✓
	Cadena Original Sello	
	Fecha Timbrado	
	No Certificado SAT	
	Sello CDF	
	Sello SAT	
	UUID	
	Tiempo de proceso	
	Archivo PDF	✓
	Código de error	✓
Mensaje de error	✓	

URL DE ACCESO AL SERVICIO

<http://www.factorumweb.com/wsportalv32/>

Se necesita firmar para que pueda entrar al portal:



Portal para el Web Service

Usuario:

R.F.C.:

Clave :

Una vez firmado, puede seleccionar entre dos opciones, una para pruebas y otra para productivo, una vez que proporcione su logo, puede aparecer en la pantalla:



NO LOGO AVAILABLE

En cualquiera de las dos opciones, deberá subir un XML ya sellado para poder timbrarlo:



NO LOGO AVAILABLE

Buscar el archivo XML que se timbrará

Cuando se sube el archivo y se timbra, nos sale una pantalla para poder recoger la información

NO LOGO AVAILABLE

Regresar a timbrar otro archivo

←

Obtener Archivo XML Obtener Archivo GIF Obtener Archivo PDF

PROPIEDAD DE FACTORUM

FAQ

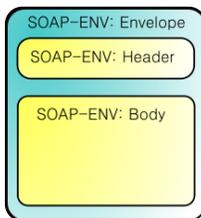
¿QUÉ ES UN WEB SERVICE?

Los web services permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. Todo esto mediante protocolos estándares

¿PARA QUÉ ME SIRVE EL WDSL?

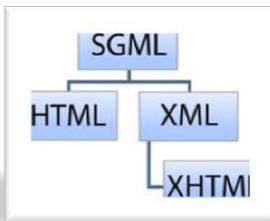
WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir Web Services. Estos sirven como documentación técnica y por lo tanto como referencia del web service.

¿QUÉ SIGNIFICA ESTAR EN SOAP?



SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Es uno de los protocolos utilizados en los servicios Web.

¿POR QUÉ USAR XML?



XML es un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Aparte de ser una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

Es el formato de entrada que recibe el SAT para efectuar el Timbre.

¿QUÉ ES EL TIMBRE EN MI FACTURA?



Se refiere al folio fiscal identificador único por cada factura generada, que genera el SAT por cada petición que se le hace.

¿QUÉ INFORMACIÓN HAY EN EL CÓDIGO BIDIMENSIONAL?



Información del timbre generado por el SAT (UUID), el RFC emisor y receptor, así como el total de la factura.

QUE ES EL UUID REGRESADO POR EL SAT?

UUID son las siglas en inglés del Identificador Universalmente Único. Esto es un código identificador estándar. El UUID del Sistema de Administración Tributaria es el equivalente al Folio que antes se otorgaba a los contribuyentes, la diferencia es que ahora no se tiene

que solicitar al SAT, sino que los Proveedores Autorizados de Certificación (PAC) lo asignan al momento de realizar la validación del documento.

El código se forma por una cadena compuesta por 32 dígitos hexadecimales (del 0 al 9 y las primeras 6 letras del alfabeto) mostrados en 5 grupos separados por guiones de la forma (8 – 4 – 4 – 4 – 12).

¿QUÉ ES EL SELLO CFD?

¿QUÉ ES LA CADENA ORIGINAL?

¿QUÉ ES EL NO. DE CERTIFICADO SAT?

¿QUÉ ES EL SELLO SAT?

¿QUIÉN PUEDE USAR EL WEB SERVICE DE FACTORUM?

Todos que generen un XML ya sellado y validado.

¿EN QUÉ TECNOLOGÍA PUEDO USAR EL WEB SERVICE?

Actualmente casi todos los lenguajes de programación soportan el consumo de web services, entre algunos tenemos: PHP, Java, C#, VB.NET



¿LA GENERACIÓN DEL TIMBRE MEDIANTE EL CONSUMO DEL WEB SERVICE ES SEGURO?



Todas las transacciones son seguras porque contamos con HTTPS,(Hypertext Transfer Protocol Secure) es una combinación del protocolo HTTP y protocolos criptográficos que se emplean para lograr conexiones más seguras

¿CUÁNTO TIEMPO TARDA EN RESPONDER EL WEB SERVICE?

El tiempo de respuesta es en general de inmediato, pero dependemos de los servidores y la comunicación del SAT.

¿CÓMO PUEDO USARLO PARA PROBARLO Y VER SI ME SIRVE?

Vía ticket puedes solicitar el usuario de prueba y bajar los ejemplos que tenemos. Cuando es usuario de prueba entonces no se guardará nada en nuestras bases de datos, y sólo se podrán usar métodos de pruebas (TEST) dentro del web service.

¿CÓMO INTEGRO EL WEB SERVICE EN MI SISTEMA O PROGRAMACIÓN?

El web service se consume como cualquier otro web service, ya que manejamos estándares en su uso e intercambio de información, se puede basar en el ejemplo que ponemos para el consumo del mismo.

Prácticamente es poner una referencia al web service, hacer la instancia correspondiente, mandar llamar el método adecuado (ver WSDL), y capturar el resultado en la clase de regreso.

En el manual técnico se detallan los métodos, sus parámetros, y la variable de retorno.

¿QUÉ VENTAJAS TENGO AL CONTRATAR EL WEB SERVICE?

XML con nodo integrado del timbre en formato UTF-8

Código bidimensional

Actualización de reglas de negocio en caso de que el SAT realice algún cambio

Servicio 24 horas 365 días del año

Servicio de soporte por sistema de tickets

Servicio transacciones seguras por https (Verisign)