

# API de Timbrado CFDI

**Manual de Usuario**



## Contenido

---

1.0 ¿Cómo timbrar con EdifactMx?	03
2.0 ¿Cómo lo envío a EdifactMx para que lo timbre?	04
3.0 Request	05
4.0 Response	06
5.0 Errores	07
6.0 Código Exitoso	08
7.0 Timbre Fiscal Digital	08
8.0 CFDI Timbrado	09
9.0 Implementación en PHP	10
10.0 Implementación en JAVA	11
11.0 Implementación en Asp.Net	13
12.0 ¿Como se realiza la cancelación en EdifactMx?	14
13.0 ¿Cómo envío la cancelación a EdifactMx?	15
14.0 Request de cancelación	16
15.0 Response de cancelación	17
16.0 Errores	18
17.0 Código Exitoso	18
18.0 Acuses de la cancelación	19
19.0 Implementación en PHP	20
20.0 Implementación en JAVA	21
21.0 Implementación en Asp.Net	22



Todos los archivos mostrados a lo largo de este manual se encuentran en la carpeta EdifactMx kit.

## 2.0 ¿Cómo lo envío a EdifactMx para que lo timbre?

El proceso para el envío a EdifactMx es mediante un web service. Existe un web service de pruebas y otro de producción, para poder hacer uso de ambos servicios es necesario estar registrado.

La dirección en la que se ubica el WSDL de pruebas es:

<http://comprobantes-fiscales.com/service/timbraCFDI.php?wsdl>

A continuación se presenta el WSDL del web service de pruebas:

```
<?xml version='1.0' encoding='utf-8'>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:respuestaTimbrado" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsi="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:respuestaTimbrado">
  <types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="urn:respuestaTimbrado">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      <xsd:complexType name="respuestaTimbrado">
        <xsd:all>
          <xsd:element name="documentoTimbrado" type="xsd:string" />
          <xsd:element name="codigoResultado" type="xsd:string" />
          <xsd:element name="codigoDescripcion" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
    </xsd:schema>
  </types>
  <message name="timbrarCFDIRequest">
    <part name="suscriptorRFC" type="xsd:string" />
    <part name="agenteFI" type="xsd:string" />
    <part name="documentoXML" type="xsd:string" />
  </message>
  <message name="timbrarCFDIResponse">
    <part name="return" type="tns:respuestaTimbrado" />
  </message>
  <portType name="timbrarCFDIPortType">
    <operation name="timbrarCFDI">
      <input message="tns:timbrarCFDIRequest" />
      <output message="tns:timbrarCFDIResponse" />
    </operation>
  </portType>
  <binding name="timbrarCFDIBinding" type="tns:timbrarCFDIPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="timbrarCFDI">
      <soap:operation soapAction="http://comprobantes-fiscales.com/service/timbrarCFDI.php/timbrarCFDI" style="rpc" />
      <input>
        <soap:body use="encoded" namespace="urn:respuestaTimbrado" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:respuestaTimbrado" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="timbrarCFDI">
    <port name="timbrarCFDIPort" binding="tns:timbrarCFDIBinding">
      <soap:address location="http://comprobantes-fiscales.com/service/timbrarCFDI.php" />
    </port>
  </service>
</definitions>
```

La dirección del web service de producción se dará una vez terminado el proceso de pruebas, cubriendo un mínimo de timbrados exitosos, para asegurar que la aplicación emisora del CFDI está lista para timbrar con EdifactMx.

[illegible]



[illegible]

## 5.0 Errores

Sí en el proceso de timbrado el sistema detecta que el documento (CFDI sin timbrar) contiene errores, devolverá alguno de los siguientes códigos:

- 101 - Suscriptor inactivo.
- 102 - Suscriptor no registrado.
- 103 - Agente TI no registrado.
- 104 - El RFC es inválido.
- 105 - El certificado no se encuentra en la LCO.
- 106 - Emisor sin validez de obligaciones.
- 301 - XML mal formado, esquema inválido.
- 302 - El sello del emisor es inválido.
- 303 - Sello no corresponde a emisor.
- 304 - Certificado Revocado o caduco.
- 305 - La fecha de emisión no esta dentro de la vigencia del CSD del Emisor.
- 306 - EL certificado no es de tipo CSD.
- 307 - El CFDI contiene un timbre previo.
- 308 - Certificado no expedido por el SAT.
- 401 - Fecha y hora de generación fuera de rango, debe ser menor a 72 horas.
- 402 - RFC del emisor no se encuentra en el régimen de contribuyentes.
- 403 - La fecha de emisión no es posterior al 01 de enero 2011.
- 000 - Error Interno.

Es importante mencionar no todos estos códigos de error serán devueltos en el ambiente de pruebas debido a que no se hacen todas las validaciones de la Lista de Contribuyentes Obligados (LCO).



## 6.0 Código Exitoso

Sí en el proceso de timbrado no se detectó ningún inconveniente el sistema devolverá el siguiente código:

- 100 - XML timbrado correctamente.

## 7.0 Timbre Fiscal Digital

Cuando la respuesta fue exitosa devuelve el documento más el timbre fiscal digital como complemento en el XML.

A continuación se muestra el timbre fiscal digital:

```
<cfel:Complemento>
  <tfd:TimbreFiscalDigital xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.sat.gob.mx/TimbreFiscalDigital http://www.sat.gob.mx/xitiu_internet/TimbreFiscalDigital/TimbreFiscalDigital.xsd" FechaTimbrado="2017-02-17T11:27:36" UUID="159048a6-7d55-4387-bd2a-ba048a2641d9" noCertificadoSAT="20001000000300022323" selloCFD="99vPv735qX0DL/K00H9+7d5ALZ7FyCnT0VaZ0fyRg2g3n8p0a1D18A2D9CWzDRI+cxGEdoCTiWQ9d0t32K/PtCk3-UY0R0UG94z0v6qK7Tq7T07p4KoyWOCz/fy5w2pJpPt48pK4htoschRanf0BupHh1M7537tw1AT=" selloSAT="M6vCyd1et3hA1lv6ryypt1ntstHwI8r0y34Cg0V/bcnfcm0a02ut0LVSYS0u0e1v0mgp0p121LDP1LrwVg51eF0043u3Xq0a009a4q08P0e1345351f1jReK1B9FX1V5C8pud00x00v0P000BLA1+H172c190C0u0vP071a04y/AonfCk1LB40P0w0+eYn2jVU0P0h1q051300006jy0nt0y/Http0s/00d082p0aCjL0Hv20pCvg2em0L0mTo0j10p51BL07F0j35/rFw0B15P0Itx57j1nFM0Qw0u0Kqod5u0000y0sh0WATH0AF0uCs0u0=" version="1.0"></tfd:TimbreFiscalDigital>
</cfel:Complemento>
```

Como se puede observar la etiqueta TimbreFiscalDigital contiene los siguientes atributos:

- FechaTimbrado - Es la Fecha y hora que se timbro el documento.
- UUID - Es el identificador único universal.
- noCertificadoSAT - Es el numero del certificado de sello digital del PAC.
- selloCFD - Es el sello que el emisor generó antes del que se enviara a timbrar el documento.
- selloSAT - Es el sello que el PAC genero con el documento que el emisor envió.
- Versión - Versión del timbre.



Como podemos observar el Timbre Fiscal Digital se coloca en la parte inferior como un complemento, con las características que se comentaron anteriormente.

## 9.0 Implementación en PHP

Para el consumo del servicio web de timbrado en PHP se utilizó la librería “NuSoap PHP”.

A continuación un ejemplo:

```
<?php
date_default_timezone_set('America/Merida');
require_once('nusoap.php');

$file = file_get_contents("CFDI_sinTimbrar.xml");
$xml = new DOMDocument();
$xml->loadXML($file);
$wsdl = "http://comprobantes-fiscales.com/service/timbrarCFDI.php?wsdl";
$client = new nusoap_client($wsdl, true);
$client->soap_defencoding = 'UTF-8';
$client->setDebugLevel(0);
$xml->saveXML($xml);
$response = $client->call('timbrarCFDI', array('suscriptorRFC' => "EDI101020E99", 'agenteTI' => "EDI101020E99", 'documentoXML' => $xml));

echo "<?Request</?><pre>";
echo "<?Request</?><pre>";
echo "<pre>";
print_r($response);
echo "</pre>";
?>
```

El código mostrado anteriormente muestra en la pantalla del navegador lo siguiente:

### Request

```
POST /service/comprobantes-fiscales.com HTTP/1.0
Host: comprobantes-fiscales.com
User-Agent: NuSOAP/0.9.5 (1.123)
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://comprobantes-fiscales.com/service/comprobantes-fiscales.com/EnviarNuevoComprobante"
Content-Length: 4422

<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

### Response

```
HTTP/1.1 200 OK
Date: Mon, 28 Feb 2017 17:30:03 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips mod_vhost_lisp/2.4.0
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Content-Length: 2544
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es para fines demostrativos por lo que al realizar el consumo deberán ser reemplazada por información del cliente.

## 10.0 Implementación en JAVA

A continuación un ejemplo:

```
package soapmanual;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import javax.xml.soap.MessageFactory;
import javax.xml.soap.MimeHeaders;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPBodyElement;
import javax.xml.soap.SOAPConnection;
import javax.xml.soap.SOAPConnectionFactory;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPMessage;
import javax.xml.soap.SOAPPart;

public class SoapManual {
    public static void main(String[] args) throws FileNotFoundException, IOException, SOAPException, Exception {
        // Create SOAP Connection
        SOAPConnectionFactory soapConnectionFactory = SOAPConnectionFactory.newInstance();
        SOAPConnection soapConnection = soapConnectionFactory.createConnection();
        String url = "http://comprobantes-fiscales.com/service/timbrarCFDI.php?wsdl";
        SOAPMessage soapResponse = soapConnection.call(createSOAPRequest(), url);
        SOAPBody responseBody = soapResponse.getSOAPBody();
        SOAPBodyElement responseElement = (SOAPBodyElement) responseBody.getChildElements().next();
        SOAPElement returnElement = (SOAPElement) responseElement.getChildElements().next();
        String respuesta = getXmlFromSOAPMessage(soapResponse);
        System.out.println(respuesta);
        soapConnection.close();
    }

    private static String getXmlFromSOAPMessage(SOAPMessage message) throws SOAPException, IOException {
        ByteArrayOutputStream byteArrayOS = new ByteArrayOutputStream();
        message.writeTo(byteArrayOS);
        return new String(byteArrayOS.toByteArray());
    }

    private static SOAPMessage createSOAPRequest() throws Exception {
        MessageFactory messageFactory = MessageFactory.newInstance();
        SOAPMessage soapMessage = messageFactory.createMessage();
        SOAPPart soapPart = soapMessage.getSOAPPart();
        //
        File file = new File("/Users/andreschable/Sites/registrovalidapp/CFDI_sintimbrar.xml");
        byte[] bytesArray = new byte[(int) file.length()];
        try (FileInputStream fis = new FileInputStream(file)) {
            fis.read(bytesArray);
        }
        String string = new String(bytesArray, "UTF-8");
        String s = new sun.misc.BASE64Encoder().encode(string.getBytes("UTF-8"));
        SOAPEnvelope se = soapPart.getEnvelope();
        se.removeNamespaceDeclaration(se.getPrefix());
        se.addNamespaceDeclaration("soapenv", "http://www.w3.org/2003/12/soap-envelope");
        se.setPrefix("soapenv");
        soapPart.getEnvelope().getHeader().setPrefix("soapenv");
        soapPart.getEnvelope().getBody().setPrefix("soapenv");

        SOAPBody soapBody = se.getBody();
        SOAPElement soapBodyElem = soapBody.addChildElement("timbrarCFDI");
        SOAPElement soapBodyElem1 = soapBodyElem.addChildElement("suscriptorRFC");
        soapBodyElem1.addTextNode("EDI101020E99");
        SOAPElement soapBodyElem2 = soapBodyElem.addChildElement("agenteTI");
        soapBodyElem2.addTextNode("EDI101020E99");
        SOAPElement soapBodyElem3 = soapBodyElem.addChildElement("documentoXML");
        soapBodyElem3.addTextNode(s);
        MimeHeaders headers = soapMessage.getMimeHeaders();
        headers.addHeader("SOAPAction", "http://comprobantes-fiscales.com/service/timbrarCFDI.php/timbrarCFDI");
        soapMessage.saveChanges();
        return soapMessage;
    }
}
```

NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es para fines demostrativos por lo que al realizar el consumo deberán ser reemplazada por información del cliente.

## 11.0 Implementación en Asp.Net

A continuación un ejemplo:

```
using System;using System.Collections.Generic;using System.Linq;
using System.Web;using System.Web.Mvc;using System.Web.Mvc.Ajax;
using System.IO;using System.Text;using System.Net;using System.Xml;
namespace soapT.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            String suscriptor = "EDI101020E99";
            String agente = "EDI101020E99";
            String xmlBase64 = "PD94bWwgdGVyc2lvb0j954wTjBibmVZQ1uZz0jVVRGLTgiPz48Y2ZkaT06b21wcm91YV50ZS84bWkuczpjZmR0PS3odiHw0iBvd3d3Lrnhvd
            String empaquetado = "<soapenv:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:xsd='http://www.w3.org/200
            "<suscriptorRFC xsi:type='xsd:string'>"+suscriptor+"</suscriptorRFC> " +
            "<agenteTI xsi:type='xsd:string'>"+agente+"</agenteTI> " +
            "<documentoXML xsi:type='xsd:string'>"+xmlBase64+"</documentoXML> " +
            "</urn:timbrarCFDI></soapenv:Body></soapenv:Envelope>";
            String urlServicio = "http://comprobantes-fiscales.com/service/timbrarCFDI.php?wsdl";
            String urlAction = "http://comprobantes-fiscales.com/service/timbrarCFDI.php/timbrarCFDI";
            ViewData["respuestaWS"] = ATask(urlServicio,urlAction,empaquetado);
            return View();
        }
        public static string ATask(string uri, string soapOperation, string xmlRequestText)
        {
            string xmlResponseText = "";
            HttpWebRequest httpRequest = (HttpWebRequest)WebRequest.Create(uri);
            httpRequest.Headers.Add("SOAPAction", soapOperation);
            httpRequest.ContentType = "text/xml; charset=utf-8";
            httpRequest.Method = "POST";
            XmlDocument xmlRequestDocument = new XmlDocument();
            xmlRequestDocument.LoadXml(xmlRequestText);
            using (Stream stream = httpRequest.GetRequestStream())
            {
                xmlRequestDocument.Save(stream);
            }
            using (WebResponse response = httpRequest.GetResponse())
            {
                using (StreamReader reader = new StreamReader(response.GetResponseStream()))
                {
                    xmlResponseText = reader.ReadToEnd();
                }
            }
            return xmlResponseText;
        }
    }
}
```

Al ejecutar el código anterior se abrirá el navegador y mostrará la siguiente respuesta:

Response:

```
<?xml version='1.0' encoding='UTF-8'?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding' xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding' xmlns:tns='urn:respuestaTimbrado'>
<SOAP-ENV:Body><tns:timbrarCFDIResponse xmlns:xsi='urn:respuestaTimbrado'><cream xsi:type='tns:respuestaTimbrado'><documentoTimbrado
xsi:type='xsd:string'>PD94bWwgdGVyc2lvb0j954wTjBibmVZQ1uZz0jVVRGLTgiPz48Y2ZkaT06b21wcm91YV50ZS84bWkuczpjZmR0PS3odiHw0iBvd3d3Lrnhvd
</documentoTimbrado><codigoResultado xsi:type='xsd:string'>100</codigoResultado><codigoDescripcion xsi:type='xsd:string'>CFDI timbrado correctamente.</codigoDescripcion></cream>
</tns:timbrarCFDIResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es para fines demostrativos por lo que al realizar el consumo deberán ser reemplazada por información del cliente.



Para comenzar el proceso de una cancelación es necesario tener el xml de cancelación, como está definido en el anexo 20 del SAT.

A continuación se muestra un ejemplo de un xml de cancelación:

Después de tener xml de cancelación se firma utilizando el Certificado de Sello Digital del emisor del CFDI, bajo el estándar XML Digital Signature establecido por el W3C (<http://www.w3.org/TR/xmlsigcore>) identificando cada uno de los CFDI a cancelar por medio del identificador único incluido en el Timbre Fiscal Digital.

A continuación se muestra un ejemplo del xml firmado digitalmente:

[illegible]



## 13.0 ¿Cómo envío la cancelación a EdifactMx?

La cancelación del CFDI también se realiza mediante un web service. Existe un web service de pruebas y otro de producción.

La dirección en la que se ubica el WSDL de pruebas es:

<http://comprobantes-fiscales.com/service/cancelarCFDI.php?wsdl>

A continuación se presenta el WSDL del web service de pruebas.

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://schemas.xmlsoap.org/soap/envelope/">
  <types>
    <xsd:schema xmlns:tns="http://schemas.xmlsoap.org/soap/envelope/">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/envelope/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/envelope/" />
    </xsd:schema>
  </types>
  <message name="enviarMensajeCancelacionRequest">
    <part name="mensaje" type="xsd:string"/>
  </message>
  <message name="enviarMensajeCancelacionResponse">
    <part name="mensaje" type="xsd:string"/>
  </message>
  <portType name="enviarMensajeCancelacionPort">
    <operation name="enviarMensajeCancelacion">
      <input message="enviarMensajeCancelacionRequest"/>
      <output message="enviarMensajeCancelacionResponse"/>
    </operation>
  </portType>
  <binding name="enviarMensajeCancelacionBinding" type="tns:enviarMensajeCancelacionPort">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="enviarMensajeCancelacion">
      <soap:operation soapAction="http://schemas.xmlsoap.org/soap/envelope:enviarMensajeCancelacion" />
      <input>
        <soap:body use="encoded" namespace="http://schemas.xmlsoap.org/soap/envelope/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="http://schemas.xmlsoap.org/soap/envelope/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="enviarMensajeCancelacion">
    <port name="enviarMensajeCancelacionPort" binding="tns:enviarMensajeCancelacionBinding">
      <soap:address location="http://comprobantes-fiscales.com/service/cancelarCFDI.php" />
    </port>
  </service>
</definitions>
```

La dirección del web service de producción se dará una vez terminado el proceso de pruebas.

[illegible]

[illegible]

## 16.0 Errores

Si en el proceso de cancelación el sistema detecta que el documento contiene errores, devolverá alguno de los siguientes códigos:

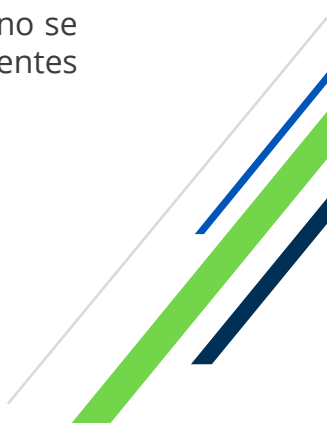
- 202 - UUID Previamente cancelado.
- 203 - UUID No corresponde al emisor.
- 204 - UUID No aplicable para cancelación.
- 205 - UUID No existe.
  
- 301 - XML mal formado, esquema inválido.
- 302 - El sello del emisor es inválido.
- 303 - Sello no corresponde a emisor.
- 304 - Certificado Revocado o caduco.
- 305 - La fecha de emisión no esta dentro de la vigencia del CSD del Emisor.
- 306 - EL certificado no es de tipo CSD.
- 307 - El CFDI contiene un timbre previo.
- 308 - Certificado no expedido por el SAT.
  
- 401 - Fecha y hora de generación fuera de rango, debe ser menor a 72 horas.
- 402 - RFC del emisor no se encuentra en el régimen de contribuyentes.
- 403 - La fecha de emisión no es posterior al 01 de enero 2011.
  
- 000 - Error Interno.

## 17.0 Código Exitoso

- 201 - UUID Cancelado.

Como se puede observar los códigos 202 al 205 hacen referencia al UUID y los códigos del 301 al 308 y del 401 al 403 hacen referencia a la petición (request).

Es importante mencionar que no todos estos códigos de error serán devueltos en el ambiente de pruebas debido a que no se hacen todas las validaciones de la Lista de Contribuyentes Obligados(LCO).



Existen 2 diferentes tipos de acuses:

A continuación se muestra un ejemplo de acuse con el **código de error 302** - El sello del emisor es inválido.

2.- Cuando no existe error en la petición. Una vez que la petición fue correcta, y se pasaron las validaciones correspondientes a los datos del contribuyente, se procede a hacer la cancelación de cada uno de los UUID's indicados, regresando el estatus de cada uno como se corresponda con los registros del SAT. Si la cancelación es exitosa regresa el acuse con **<EstatusUUID> 201</EstatusUUID>**.

Si no es así se regresa algún código del 202 al 205 .



NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es para fines demostrativos por lo que al realizar el consumo deberán ser reemplazada por información del cliente.



## 20.0 Implementación en JAVA

A continuación un ejemplo:

```
package soapmanual;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import javax.xml.soap.MessageFactory;
import javax.xml.soap.MimeHeaders;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPBodyElement;
import javax.xml.soap.SOAPConnection;
import javax.xml.soap.SOAPConnectionFactory;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPMessage;
import javax.xml.soap.SOAPPart;

public class SoapManual {
    public static void main(String[] args) throws FileNotFoundException, IOException, SOAPException, Exception {
        // Create SOAP Connection
        SOAPConnectionFactory soapConnectionFactory = SOAPConnectionFactory.newInstance();
        SOAPConnection soapConnection = soapConnectionFactory.createConnection();
        String url = "http://comprobantes=fiscales.com/service/cancelarCFDI.php?wsdl";
        SOAPMessage soapResponse = soapConnection.call(createSOAPRequest(), url);
        SOAPBody responseBody=soapResponse.getSOAPBody();
        SOAPBodyElement responseElement= (SOAPBodyElement) responseBody.getChildElements().next();
        SOAPElement returnElement=(SOAPElement)responseElement.getChildElements().next();
        String respuesta= getXmlFromSOAPMessage(soapResponse);
        System.out.println(respuesta);
        soapConnection.close();
    }

    private static String getXmlFromSOAPMessage(SOAPMessage message) throws SOAPException, IOException {
        ByteArrayOutputStream byteArrayOS= new ByteArrayOutputStream();
        message.writeTo(byteArrayOS);
        return new String(byteArrayOS.toByteArray());
    }
}
```

Al ejecutar la clase mostrada con anterioridad, en la ventana de salida nos mostrará lo siguiente:

En este ejemplo se marca el código que nos indica que la cancelación fue correcta.

```
run:
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-E
NS="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XM
LSchema-instance" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:enviaAcuseCancelacionRe
sponse xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"><ns1:return xsi:type="xsd:string"><?xml version="1.0" encoding="
UTF-8"?><Acuse xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Fecha="2
017-02-20T17:58:59" RfcEmisor="EDI101020E99"><Folios xmlns="http://cancelacfd.sat.gob.mx"><UID6gt;159b40a6-7d55-458
7-8d2a-bab48a2641d96lt;/UID6gt;<EstatusUID6gt;2016lt;/EstatusUID6gt;<Folios6gt;<Signature xmlns="http://www.w3.org/
2000/09/xmldsig#" Id="SelloSAT"><SignedInfo6gt;<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xm
l-20010315"/><SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha512"/><Reference URI=""><
lt;Transforms6gt;<Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116"><XPath6gt;not(ancestor-or-self::i
local-name()='Signature'))</XPath6gt;<Transform6gt;<DigestMethod Algorithm="http://www.w3.org/2001/
04/xmldsig#sha512"/><DigestValue6gt;2V5arlnfAw3pQUP+2vbnY2f3b5fYe+71hIDW/vkotrAFzm/su0w9CMYADUg027W435zcR3vFak6b4qao/FI/Qe
=6lt;/DigestValue6gt;<Reference6gt;<SignedInfo6gt;<SignatureValue6gt;23m0FiulNgXFESudF1+XULc380bWdVx/wTqATxL0i4hxk
Vbqd/D5CIm=8zXINXev7gV8NpLWqVFMH9H9d9=6lt;/SignatureValue6gt;<KeyInfo6gt;<KeyName6gt;00001088888800000936lt;/KeyName6gt
<KeyValues6gt;<RSAPublicKey6gt;<Modulus6gt;yxMvUucU+s3aewTFZv3rrFWIdes7kIDJm070A5DP+ZTapoFnt37fgeIHL1TUDAVvd/fDKhfiwN5H=
vbrN058X3UEd0Qr3ngb6zp/rDjgYsedckPLv6fro4D0BNXLdALFqHn8ARyX77kY8NvI1f05Vp401Vc3urLUTiEm16Kle3t0ywhfjgFzdk3oAIXFBoeeL/GburWbJn
pP=NeGahV5bKxLCp575nKVonXwzpf6GuBp204NGk12/jYA2EH8wyR94yUvzjT7I3YrHng23kLRDlJorYwa9BQOPdQ5TpcrUu8nLhpQdI/zMTLONF2NiBCKNuAM
acKhnvLWm=6lt;/Modulus6gt;<Exponent6gt;AQAB6lt;/Exponent6gt;<RSAPrivateKey6gt;<KeyValues6gt;<KeyInfo6gt;<Signature6gt;<Acuse6gt;<ns1:return><ns1:enviaAcuseCancelacionResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
BUILD SUCCESSFUL (total time: 1 second)
```

NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es para fines demostrativos por lo que al realizar el consumo deberán ser reemplazada por información del cliente.

## 21.0 Implementación en Asp.Net

A continuación un ejemplo:

```
using System;using System.Collections.Generic;using System.Linq;
using System.Web;using System.Web.Mvc;using System.Web.Mvc.Ajax;
using System.IO;using System.Text;using System.Net;using System.Xml;
namespace soapT.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            String xml = "<?xml version='1.0' encoding='UTF-8'><?xml:namespace prefix='http://schemas.xmlsoap.org/soap/envelope/' xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'><soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'><soap:Body><ns2:enviaAcuseCancelacion xmlns='http://dom.w3c.org/xsd' xmlns:ns2='http://edifact.com.mx/xsd'><ns2:mensajeXsd><ns2:mensajeXsd></ns2:mensajeXsd></ns2:enviaAcuseCancelacion></soap:Body></soap:Envelope>";
            String urlServicio = "http://comprobantes-fiscales.com/service/cancelarCFDI.php?wsdl";
            String urlAction = "http://comprobantes-fiscales.com/service/cancelarCFDI.php/enviaAcuseCancelacion";
            ViewData["requestXML"] = ATask(urlServicio, urlAction, empaquetado);
            return View();
        }

        public static string ATask(string uri, string soapOperation, string xmlRequestText)
        {
            string xmlResponseText = "";
            HttpWebRequest httpRequest = (HttpWebRequest)WebRequest.Create(uri);
            httpRequest.Headers.Add("SOAPAction", soapOperation);
            httpRequest.ContentType = "text/xml; charset=utf-8";
            httpRequest.Method = "POST";
            XmlDocument xmlRequestDocument = new XmlDocument();
            xmlRequestDocument.LoadXml(xmlRequestText);
            using (Stream stream = httpRequest.GetRequestStream())
            {
                xmlRequestDocument.Save(stream);
            }
            using (WebResponse response = httpRequest.GetResponse())
            {
                using (StreamReader reader = new StreamReader(response.GetResponseStream()))
                {
                    xmlResponseText = reader.ReadToEnd();
                }
            }
            return xmlResponseText;
        }
    }
}
```

Al ejecutar el código anterior se abrirá el navegador y mostrará la siguiente respuesta:

En este ejemplo se marca el código que nos indica que la cancelación fue correcta.

Response:

```
<?xml version='1.0' encoding='UTF-8'><SOAP-ENV:Envelope SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'><SOAP-ENV:Body>
<ns1:enviaAcuseCancelacionResponse xmlns='http://edifact.com.mx/sod' xsi:type='soap:body'><ns1:mensajeXsd><ns1:mensajeXsd></ns1:mensajeXsd></ns1:enviaAcuseCancelacionResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

NOTA: Con excepción de las direcciones de servicios web, la información incluida en los códigos de implementación es



+ información

*[www.edifact.com.mx](http://www.edifact.com.mx)*

01 800 FACTURA

