

Proyecto: Integración de Timbrado usando Visual Fox Pro 9.0

Documento: Manual de implementación

Definiciones de métodos

Es necesario agregar las definiciones de los métodos que vamos a consumir, ya que se van a enviar con los parámetros al servicio web, estas definiciones las tomamos de lo mostrado en el explorador, al seleccionar cada uno de los métodos.

Timbrado

Click [here](#) for a complete list of operations.

TimbraCFDI

Servicio de timbrado para Integradores

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /servicioIntegracionpruebas/timbrado.asmx HTTP/1.1
Host: 201.178.12.89
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/TimbraCFDI"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TimbraCFDI xmlns="http://localhost/">
      <usuarioIntegrador>string</usuarioIntegrador>
      <xmlComprobanteBase64>string</xmlComprobanteBase64>
      <idComprobante>string</idComprobante>
    </TimbraCFDI>
  </soap:Body>
</soap:Envelope>
```

METODO Registra Emisor

&&Xml cabecera Registra Emisor

Public strRegistraEmisor as Variant

strRegistraEmisor = '<?xml version="1.0" encoding="utf-8"?>'

strRegistraEmisor = strRegistraEmisor+ '<soap:Envelope

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">'

strRegistraEmisor = strRegistraEmisor+ '<soap:Body>'

*- Nombre del Método

strRegistraEmisor = strRegistraEmisor+ '<RegistraEmisor

xmlns="http://localhost/">'

*- Nombre de la propiedad pasada como parámetro

strRegistraEmisor = strRegistraEmisor+

'<usuarioIntegrador>string</usuarioIntegrador>'

strRegistraEmisor = strRegistraEmisor+ '<rfcEmisor>string</rfcEmisor>'

strRegistraEmisor = strRegistraEmisor+ '<base64Cer>string</base64Cer>'

strRegistraEmisor = strRegistraEmisor+ '<base64Key>string</base64Key>'

strRegistraEmisor = strRegistraEmisor+ '<contrasena>string</contrasena>'

```
strRegistraEmisor = strRegistraEmisor+ '</RegistraEmisor>'  
strRegistraEmisor = strRegistraEmisor+ '</soap:Body>'  
strRegistraEmisor = strRegistraEmisor+ '</soap:Envelope>'
```

METODO TimbraCFDI

```
Public strTimbraCFDI as Variant  
strTimbraCFDI='<?xml version="1.0" encoding="utf-8"?>'  
strTimbraCFDI=strTimbraCFDI + '<soap:Envelope  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">'  
strTimbraCFDI= strTimbraCFDI + '<soap:Body>'  
*- Nombre del Método  
strTimbraCFDI= strTimbraCFDI + '<TimbraCFDI xmlns="http://localhost/">'  
*- Nombre de la propiedad pasada como parámetro  
strTimbraCFDI= strTimbraCFDI +  
'<usuarioIntegrador>string</usuarioIntegrador>'  
strTimbraCFDI= strTimbraCFDI +  
'<xmlComprobanteBase64>string</xmlComprobanteBase64>'  
strTimbraCFDI= strTimbraCFDI + '<idComprobante>string</idComprobante>'  
strTimbraCFDI= strTimbraCFDI + '</TimbraCFDI>'  
strTimbraCFDI= strTimbraCFDI + '</soap:Body>'  
strTimbraCFDI= strTimbraCFDI + '</soap:Envelope>'
```

METODO CancelaCFDI

```
Public strCancelaCFDI as Variant  
strCancelaCFDI='<?xml version="1.0" encoding="utf-8"?>'  
strCancelaCFDI= strCancelaCFDI + '<soap:Envelope  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">'  
strCancelaCFDI= strCancelaCFDI + '<soap:Body>'  
*- Nombre del Método  
strCancelaCFDI= strCancelaCFDI + '<CancelaCFDI xmlns="http://localhost/">'  
*- Nombre de la propiedad pasada como parámetro  
strCancelaCFDI= strCancelaCFDI +  
'<usuarioIntegrador>string</usuarioIntegrador>'  
strCancelaCFDI= strCancelaCFDI + '<rfcEmisor>string</rfcEmisor>'  
strCancelaCFDI= strCancelaCFDI + '<folioUUID>string</folioUUID>'  
strCancelaCFDI= strCancelaCFDI + '</CancelaCFDI>'  
strCancelaCFDI= strCancelaCFDI + '</soap:Body>'  
strCancelaCFDI= strCancelaCFDI + '</soap:Envelope>'
```

'METODO ObtieneCFDI

```
Public strObtieneCFDI as Variant
strObtieneCFDI='<?xml version="1.0" encoding="utf-8"?>'
strObtieneCFDI=strObtieneCFDI + '<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">'
strObtieneCFDI= strObtieneCFDI + '<soap:Body>'
*- Nombre del Método
strObtieneCFDI= strObtieneCFDI + '<ObtieneCFDI xmlns="http://localhost/">'
*- Nombre de la propiedad pasada como parámetro
strObtieneCFDI= strObtieneCFDI +
'<usuarioIntegrador>string</usuarioIntegrador>'
strObtieneCFDI= strObtieneCFDI + '<rfcEmisor>string</rfcEmisor>'
strObtieneCFDI= strObtieneCFDI + '<folioUUID>string</folioUUID>'
strObtieneCFDI= strObtieneCFDI + '</ObtieneCFDI>'
strObtieneCFDI= strObtieneCFDI + '</soap:Body>'
strObtieneCFDI= strObtieneCFDI + '</soap:Envelope>'
```

1. TimbraCFDI

CLEAR

LOCAL parser As MSXML2.DOMDocument

LOCAL parserRespuesta As DOMDocument

parser =CREATEOBJECT('MSXML2.DOMdocument')

parserRespuesta =CREATEOBJECT('MSXML2.DOMdocument')

strSOAPAction = "http://localhost/RegistraEmisor"

parser.loadXML(strRegistraEmisor)

&&'Convierte los archivos a base64

archivoCerBase64 = STRCONV(FileToStr(SUBSTR(JUSTPATH(SYS(16)),31,100) +
 "\Archivos\aaa010101aaa__csd_01.cer"),13)

&&'C:\Users\Eleu\Documents\Desarrollos\ProyectoConsumoWSVFP\FORMS\Archivos
 \aaa010101aaa__csd_01.cer"),13)

archivoKeyBase64=STRCONV(FileToStr(SUBSTR(JUSTPATH(SYS(16)),31,100)+
 "\Archivos\aaa010101aaa__csd_01.key"),13)

&&Indicar los parámetros a enviar

parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/usuarioIn
 tegrador").text = usuarioIntegrador

parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/rfcEmisor
 ").Text = rfcEmisor

```
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/base64Cer")
.Text = archivoCerBase64
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/base64Key")
.Text = archivoKeyBase64
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/contrasena")
.Text = contrasena
```

```
SET PROCEDURE TO SUBSTR(JUSTPATH(SYS(16)),31,100)+"\program2.prg"
parserRespuesta=InvokeWebService(parser.xml,strSOAPAction,strWsdL,parserRespuesta)
```

```
IF VARTYPE(parserRespuesta)!="U" then
    Local tipoExcepcion, numError, descResultado As String
    &&tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").childNodes.Item(0).Text
    &&Numero de error
    numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").childNodes.Item(1).Text
    &&Descripcion del resultado
    descResultado =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").childNodes.Item(2).Text
    thisform.resultado.value = descResultado
endif
```

2. ObtieneCFDI

```
CLEAR
LOCAL parser As MSXML2.DOMDocument
LOCAL parserRespuesta As DOMDocument
parser =CREATEOBJECT('MSXML2.DOMdocument')
parserRespuesta =CREATEOBJECT('MSXML2.DOMdocument')

Thisform.txtffiscalc.value = ""
&&Metodo del servicio web
strSOAPAction = "http://localhost/ObtieneCFDI"
&&cargar el código SOAP para OBTIENECDI
parser.loadXML(strObtieneCFDI)

&&Folio Fiscal
folioUUID = ALLTRIM(thisform.txtffiscalc.text)

&&Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/usuarioInteg
rador").text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/rfcEmisor").
Text = rfcEmisor
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/folioUUID").
Text = folioUUID

SET PROCEDURE TO SUBSTR(JUSTPATH(SYS(16)),31,100) + "\program2.prg"
parserRespuesta=InvokeWebService(parser.xml,strSOAPAction,strWsdL,parserRe
spuesta)

IF VARTYPE(parserRespuesta)!="U" then
    Local tipoExcepcion, numError, descResultado,xml,cod64 As String
    &&tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
onse/ObtieneCFDIResult").childNodes.Item(0).Text
    &&Numero de error
    numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
onse/ObtieneCFDIResult").childNodes.Item(1).Text
    &&Descripcion del resultado
    descResultado =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
```

```

onse/ObtieneCFDIResult").childNodes.Item(2).Text
    &&XML
    xml =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
onse/ObtieneCFDIResult").childNodes.Item(3).Text
    &&codigo base64 QR
    cod64 =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
onse/ObtieneCFDIResult").childNodes.Item(4).Text
    &&Cadena original SAT
    thisform.txtcodOr.value =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResp
onse/ObtieneCFDIResult").childNodes.Item(5).Text
    &&Validamos si existe error
    If numError = "0" Then
        &&Obtenemos los datos del xml regresado timbrado
        local listaNodos As MSXML2.IXMLDOMNodeList
        listaNodos =
parserRespuesta.selectNodes("/soap:Envelope/soap:Body/ObtieneCFDIResponse/
ObtieneCFDIResult/anyType")

parserRespuesta.loadXML(STRTRAN(listaNodos.Item(3).Text,"<","<"))
        parserRespuesta.loadXML(STRTRAN(listaNodos.Item(3).Text, ">","
">"))

        &&Mostramos los daros obtenidos
        thisform.txtsdigital.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:Ti
mbreFiscalDigital/@selloSAT").Text
        thisform.txtcersat.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:Ti
mbreFiscalDigital/@noCertificadoSAT").Text
        thisform.txttimsan.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:Ti
mbreFiscalDigital/@FechaTimbrado").Text
        thisform.txtfolsat.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:Ti
mbreFiscalDigital/@UUID").Text
        &&Guarda xml obtenido

        parserRespuesta.save(ALLTRIM(SUBSTR(JUSTPATH(SYS(16)),31,100) +
"\CFDI\"+ ALLTRIM(folioUUID) + "_CFDI.xml"))

        &&Función genera QR de código base64

```

```
generaQR(cod64,ALLTRIM(folioUUID) + "QR.jpg",thisform.qrO)
listaNodos.Reset
thisform.resultado.value = "Se recupero correctamente CFDI"
Else
    thisform.resultado.value = descResultado
EndIf

ENDIF
```

3. CancelaCFDI

```
CLEAR
LOCAL parser As MSXML2.DOMDocument
LOCAL parserRespuesta As DOMDocument
parser =CREATEOBJECT('MSXML2.DOMdocument')
parserRespuesta =CREATEOBJECT('MSXML2.DOMdocument')

thisForm.txtffiscalO.value = ""

&&Metodo del servicio web
strSOAPAction = "http://localhost/CancelaCFDI"

&&cargar el código SOAP para CANCELACFDI
parser.loadXML(strCancelaCFDI)

&&Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/usuarioInteg
rador").text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/rfcEmisor").
Text = rfcEmisor
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/folioUUID").
Text = ALLTRIM(thisform.txtffiscalC.value)

SET PROCEDURE TO SUBSTR(JUSTPATH(SYS(16)),31,100) + "\program2.prg"
parserRespuesta=InvokeWebService(parser.xml,strSOAPAction,strWsdL,parserRe
spuesta)

IF VARTYPE(parserRespuesta)!="U" then
    Local tipoExcepcion, numError, descResultado,xml As String
    &&tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResp
onse/CancelaCFDIResult").childNodes.Item(0).Text
```

```
&&Numero de error
numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResp
onse/CancelaCFDIResult").childNodes.Item(1).Text
&&Descripcion del resultado
descResultado =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResp
onse/CancelaCFDIResult").childNodes.Item(2).Text
&&XML
xml =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResp
onse/CancelaCFDIResult").childNodes.Item(3).Text
thisform.resultado.value = descResultado
endif
```

4. RegistraEmisor

```
CLEAR
LOCAL parser As MSXML2.DOMDocument
LOCAL parserRespuesta As DOMDocument
LOCAL xmlComprobanteBase64 as String
LOCAL idcomprobante as Integer

parser =CREATEOBJECT('MSXML2.DOMdocument')
parserRespuesta =CREATEOBJECT('MSXML2.DOMdocument')

Thisform.txtffiscalo.value = ""
Thisform.txtffiscalc.value = ""

&&Metodo del servicio web
strSOAPAction = "http://localhost/TimbraCFDI"

&&cargar el código SOAP para OBTIENECFDI
parser.loadXML(strTIMBRACFDI)

&&archivo sin timbrar de prueba
xmlComprobanteBase64 =STRCONV(FileToStr(SUBSTR(JUSTPATH(SYS(16)),31,100) +
"\Archivos\comprobanteSinTimbrar.xml"),13)

&&Para efectos de prueba utilizaremos un numero aleatorio
idcomprobante = INT(RAND() * 100000)

&&Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/usuarioIntegr
```



```
ador").text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/xmlComprobanteBase64").Text = xmlComprobanteBase64
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/idComprobante").Text = idcomprobante
```

```
SET PROCEDURE TO SUBSTR(JUSTPATH(SYS(16)),31,100) + "\program2.prg"
parserRespuesta=InvokeWebService(parser.xml,strSOAPAction,strWSDL,parserRespuesta)
```

```
IF VARTYPE(parserRespuesta)!="U" then
    Local tipoExcepcion, numError, descResultado,xml,cod64 As String
    &&tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(0).Text
    &&Numero de error
    numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(1).Text
    &&Descripcion del resultado
    descResultado =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(2).Text
    &&XML
    xml =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(3).Text
    &&codigo base64 QR
    cod64 =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(4).Text
    &&Cadena original SAT
    thisform.TxtcodOro.value =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.Item(5).Text
    &&Validamos si existe error
    If numError = "0" Then
        &&Obtenemos los datos del xml regresado timbrado
        local listaNodos As MSXML2.IXMLDOMNode
        LOCAL rutaGuardaXML as string
        listaNodos =
parserRespuesta.selectNodes("soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult/anyType")
```

```

parserRespuesta.loadXML(STRTRAN(listaNodos.Item(3).Text,"<","<"))
parserRespuesta.loadXML(STRTRAN(listaNodos.Item(3).Text, ">",">"))

  &&Mostramos los daros obtenidos
  thisform.txtsdigitalo.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@selloSAT").Text
  thisform.txtcersato.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@noCertificadoSAT").Text
  thisform.txttimsano.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@FechaTimbrado").Text
  thisform.txtfolsato.value =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@UUID").Text
  &&Guarda xml obtenido

parserRespuesta.save(ALLTRIM(SUBSTR(JUSTPATH(SYS(16)),31,100))+"\Facturas\
"+ ALLTRIM(STR(idcomprobante)) + "Timbre.xml")

  &&Función genera QR de código base64

  generaQR(cod64, allTRIM(STR(idcomprobante)) + "QR.jpg",
thisform.qr)
  listaNodos.Reset
  thisform.resultado.value = "El comprobante fue timbrado
exitosamente"
  ELSE
    thisform.resultado.value = descResultado
  EndIf

endif

```